



Apprentissage : Paradigmes, Structures et abstractions

Henry Soldano

► To cite this version:

Henry Soldano. Apprentissage : Paradigmes, Structures et abstractions. Informatique [cs]. Université Paris-Nord - Paris XIII, 2009. tel-00514160

HAL Id: tel-00514160

<https://theses.hal.science/tel-00514160>

Submitted on 1 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Laboratoire
d'Informatique de l'Université Paris-Nord

Unité de recherche UMR CNRS 7030

MÉMOIRE

Apprentissage : paradigmes, structures et abstractions

présenté pour l'obtention
de l'Habilitation à Diriger des Recherches
Université Paris Nord

Henry Soldano

présenté et soutenu publiquement le 7 décembre 2009

devant le jury composé de

Daniel Kayser	Président
Younès Bennani	Examineur
Céline Rouveirol	Examineur
Pierre Marquis	Rapporteur
Amedeo Napoli	Rapporteur
Lorenza Saitta	Rapporteur
Jean Sallantin	Rapporteur
Joël Pothier	Invité

Thomas, je dédie ce mémoire à ta maman

Il est en général recommandé de commencer par le commencement. Ce n'est toutefois pas toujours le cas. Par exemple, si vous voulez peindre un chien en vert il est plus judicieux de commencer par la queue. Il pourrait vous mordre.

W-C. Fields

Je voudrais d'abord remercier le jury, dont les membres m'ont fait l'honneur ou l'amitié de lire, entendre ou rapporter ce mémoire. Qu'ils soient assurés que leur présence n'est en rien le fruit du hasard : elle est, profondément, la marque de mon estime. Je remercie particulièrement Lorenza Saïtta qui, littéralement entre deux avions, est venue éclairer mes pas, encore hésitants, sur les chemins de l'Abstraction. J'ai rencontré Pierre Marquis il y a bien longtemps, lors de chaleureuses *Journées Francophones d'Apprentissage* où, peut être à sa manière pensive de fumer la pipe, ou son regard assuré examinant la position des boules sur la table de billard, j'ai pressenti le brillant chercheur en devenir. Je partage avec Amedeo Napoli des origines italiennes. Lors de nos épisodiques rencontres, sur le chemin d'une salle de conférence ou au retour d'un banquet, au fond d'un siège dans un car traversant la nuit, nous avons parfois parlé sciences, et toujours de Jazz. Daniel Kayser a une place à part dans ce jury, celle d'une personnalité de L'I.A. qui n'aurait jamais tout à fait enseveli le potache batailleur caché dessous.

Céline et Younès, piliers de l'équipe A3 du LIPN, m'ont, chacun à leur manière, permis de traverser ces dernières années, sans s'offusquer de mes parfois étranges trajectoires, doutes soudain, ou inexplicables absences. Leur amitié m'est précieuse. Les membres de A³, fille de Adage, forment une équipe vivante parfois pleine de bruit et de fureur, souvent entre deux crises de fou-rire, où l'on trouve tous les punching-ball humains nécessaires au travail scientifique. Trouver un mot pour chacun prendrait plus de temps qu'il ne m'en reste pour finir cette courte page. Je ne pourrais cependant fermer celle-ci sans dénoncer Dominique pour les nombreux incidents émaillant sa présence parmi nous, mais qu'importent un peu de moutarde sur votre manteau ou un extincteur agressé dans le couloir : le plaisir de construire sa légende mérite bien quelques sacrifices. Plus sérieusement, l'arrivée récente d'Erick Alphonse dans notre équipe a été un formidable accélérateur, et je regrette qu'il soit aussi peu question de nos travaux communs dans ce mémoire. Simple question de temps, bien sûr.

Mais l'équipe n'est qu'une partie du laboratoire, auquel j'appartiens depuis longtemps. Le LIPN, bien qu'il ait atteint aujourd'hui une taille respectable, et eu son lot de querelles et de crises de croissance, a gardé sous la direction de Christophe Fouqueré une vraie convivialité, comme cela a pu être remarqué récemment lors du pot de faux-départ en retraite de Gérard Plateau (à ma connaissance le risque de tomber sur un de ses bons mots au détour d'un couloir est toujours élevé) et j'y ai heureusement bénéficié d'échanges scientifiques dépassant le cadre des équipes. Un mot spécifique pour l'équipe administrative du laboratoire, sans laquelle celui-ci n'existerait pas. Je pense en particulier à Jacqueline et Brigitte qui l'ont accompagné tout au long de sa croissance, et y ont investi bien au delà de ce que l'on pouvait attendre d'elles. Un grand merci aussi à ceux qui m'ont aidé à renverser les derniers obstacles administratifs, je pense en particulier à Christine Choppy et Laure Petrucci. Francine, j'ai toujours le nez-rouge que tu m'avais offert accroché à ma lampe.

Dans la catégorie joyeux fêtards, un prix collectif pour les fantômes du laboratoire que l'on peut de temps à autre entendre entrechoquer leurs verres tard dans la nuit (non Hayat, j'ai bien réfléchi, il vaut mieux ne pas donner de noms) et une mention spéciale et privée à Seb et Sophie,

et autres joyeuses commères de l'Ecritoire.

Je passe depuis longtemps une partie de mon temps de recherche dans un non-laboratoire, l'Atelier de BioInformatique, qui a accompagné la naissance de la BioInformatique en France. Je tiens à remercier ici Monique Pagès, alors directrice de la section Physique-Chimie de l'Institut Curie, qui a permis à cette structure informelle d'exister, et à Jean Guerdoux qui nous a ensuite accueilli au département des Sciences de la Vie de l'université Paris 6. J'y ai longtemps travaillé avec Alain Viari, sur des supports variés, principalement les nappes en papier d'une pizzeria du boulevard Saint-Michel, cotoyé Alex, Catherine, Jean-Louis, Marie-Odile, puis Claudine (te souviens-tu ?) et Manue, et bien d'autres dans la fameuse Salle Holweck, ou tout en haut du petit Paviillon. Puis nous avons rejoint la rue Cuvier et son incroyable salle des fêtes, qui me dit-on résonne parfois encore des protestations des invités découvrant les commentaires de Sophie sur les photos des soirées joyeuses au cours desquelles mais je m'égare, il y a maintenant un trou et du béton rue Cuvier, et l'ABI vibre aujourd'hui au rythme des engins du chantier de désamiantage au bas de la tour 42 (ironique réponse à la question la plus importante de l'univers, mais passons) avec Eduardo et ses troupes, Joël, le tenancier, Guillaume et Isabelle, et tous les autres, voyageurs de passage à l'Auberge de BioInformatique.

Un avant-dernier mot pour celles et ceux qui m'ont accordé leur confiance lorsqu'ils s'aventuraient sur des terres vierges, Manue, Claudine, Pierre, Marc, Marie-France, Vincent, M'hammet, Dominique, Alejandro, Nahla, Mathilde, Gauvain et Maïssa. Je leur dois une grande partie du plaisir qu'offre ce métier. Nadia, post-doc o no, ti metto qui, se ti sta bene ? baci.

Un autre avant-dernier mot, ou plutôt une ellipse légère, car il y a ceux et celles dont il ne sera rien dit, et qui comptent tout autant, si la vie est meilleure avec de vrais morceaux de science dedans, douceur et amitiés lui donnent un sens. Et, Thomas, si tu pensais t'en tirer sans un mot de ton père, c'est raté. Je t'embrasse. Maman, merci pour tout, et oui, j'ai pris mon pull en laine et mon écharpe (oui, bon, pour la couleur, c'est une longue histoire). Oui, je sais ça aurait fait plaisir à papa. Ah et la *pizza rustica* soit-disant ratée a été avalée en quelques minutes (et pour tous : non, je ne connais pas la recette).

Une pause, Amal, dans cette page, pour prendre le thé en ta compagnie, une minute élégante et délicate, avant que nous penchions encore la tête pour regarder, pensifs, s'agiter nos petits agents.

Le dernier avant-dernier pour Nathalie, et un encore pour Véronique, qui m'aurait probablement arraché les yeux si je l'avais oubliée. Tu as été, paraît-il, intenable à la soutenance, mais oui, je sais avec Pierre devant et Dominique derrière tu ne pouvais pas résister. Merci à toutes les deux, pour des moments magiques. (Mais, Véronique, faut-il aussi leur dire que c'est dans le reflet d'un faible soleil d'hiver sur les huisseries d'un voilier, ou du fond de la banquette d'un salon de thé face à l'océan que sont parfois venus ces subtils déplacements, nous posant un pas plus loin sur un chemin dont on ne trouvera ici que les lignes droites ?)

Enfin, les chercheurs, comme les vampires, ne le deviennent que s'ils ont été mordus. C'est à Jean Sallantin, un matin brumeux devant la table en bois de la cuisine de sa maison familiale à Cabourg, lorsque un ensemble de variables se sont soudain agrégées pour former une chose vivante, que je dois ce destin. Aujourd'hui encore, au moment où le vert de la peinture se met à manquer, il faut bien que je m'approche pour lui en mettre une dernière touche sur le museau ...

Table des matières

1	Préambule	9
2	Structure : Extensions, Ordres partiels, et Apprentissage supervisé	11
2.1	Extensions, Ordres partiels	11
2.1.1	Extensions, Ordres partiels	11
2.1.2	Extensions et Ordres partiels	13
2.1.3	Correspondance de Galois	14
2.1.4	Treillis de Galois des extensions/ordres partiels sur $\mathcal{P}(I)$	17
2.2	Exploration d'un espace quotient en apprentissage supervisé	18
2.2.1	Langage et Apprentissage	18
2.2.2	Apprentissage dans un espace quotient	20
2.3	Fermés fréquents en fouille de données	24
3	Abstractions Extensionnelles et Intensionnelles Ordonnées	25
3.1	Abstractions extensionnelles et intensionnelles	25
3.1.1	Projection d'un treillis et abstraction correspondante	26
3.1.2	Ordre sur les abstractions	32
3.1.3	Treillis de Galois Abstraits et ordre sur les treillis de Galois abstraits	34
3.1.4	Extension et implication abstraites	35
3.1.5	Règles d'associations abstraites	36
3.1.6	Abstraction ?	37
3.1.7	Granularité	43
3.2	La représentation Alpha : abstraction, extension, treillis Alpha des extensions/ordres partiels 46	
3.3	Perspectives	49
4	Extraction de Motifs Séquentiels, Abstraits, Relationnels	51
4.1	Préambule	51
4.2	Extraction de Motifs répétés sur des objets séquentiels et géométriques	52
4.2.1	Extraction de motifs dirigée par les données	52
4.2.2	Occurrences approchées de motifs	56
4.2.3	Des motifs aux motifs relationnels	56
4.2.4	La recherche de motifs structuraux dans les protéines	57
4.2.5	Extraction d'une structure secondaire commune à une famille de fragments d'ARN	59

4.3	Perspectives	60
5	Paradigmes	61
5.1	Apprentissage à partir de données ambiguës	61
5.1.1	LEa : Apprentissage à partir d'exemples (extrêmement) ambigus	64
5.2	Apprentissage collectif	67
5.2.1	Révision collective de théorie par maintien de la consistance dans un système multi agent : SMILE;-)	68
5.2.2	Révision collective de théorie dans une société d'agents organisée en un réseau	74
5.2.3	iSMILE : révision collective dans une société d'agents individualistes . . .	75
5.2.4	Adaptation collective dans un réseau de télécommunications	76
5.2.5	Perspectives.	76
6	Conclusion ?	79

Chapitre 1

Préambule

L'ensemble des travaux présentés ici relève de l'Apprentissage Artificiel, et aborde l'apprentissage supervisé, ainsi que l'aide à la découverte (par la fouille de données). Le titre du mémoire contient les mots *paradigmes*, *structures* et *abstractions*. Ces notions traversent, de manière plus ou moins directe, ces différents travaux.

En théorie de Probabilités on associe à un univers d'individus une mesure de probabilité élémentaire, qui s'étend pour mesurer des *événements*, c'est-à-dire des ensembles d'*individus*. Si je tire un dé non pipé n fois, un individu est constitué du résultat de n jets, et la probabilité associée (c'est un cas d'équiprobabilité) est $(1/2)^n$. Plus généralement on considère d'un point de vue théorique en Apprentissage qu'un échantillon d'apprentissage (en pratique l'ensemble d'individus constituant les données) est un ensemble de variables aléatoires *i.i.d.*, c'est-à-dire indépendantes et identiquement distribuées. La distribution (uniforme dans l'exemple des n jets), souvent inconnue, reflète la *structure* de l'univers des individus. L'apprentissage non supervisé se ramène ainsi à la recherche d'un modèle de la structure présente dans les données (pourquoi sinon catégoriser?). Un cas particulier intéressant est celui des individus *chimériques* : certains individus sont de probabilité nulle, comme un mouton à 12 pattes, reflétant par là le fait que le modèle de représentation choisi permet de représenter des individus impossibles. Prenant ici un point de vue logique plutôt que probabiliste, par *structure* on entend précisément ici la matérialisation de ce que l'*Univers des instances*¹ possibles est seulement inclus dans l'ensemble des représentations possibles. En ce sens la structure est parfois partiellement connue, et prend la forme d'une *théorie du domaine* : par exemple si une contrainte spécifie qu'un objet ne peut-être à la fois rond et carré, alors une instance dont la représentation contiendrait *rond* et *carré* ne peut apparaître dans les données.

Nous nous placerons tout le long du mémoire dans le cas où un problème d'apprentissage suppose l'exploration d'un *Espace de Recherche*, constitué de représentations dite *intensionnelles* formées dans un certain langage, et que nous appellerons *énoncés* ou *hypothèses*. Le cas intéressant est celui où la structure est en grande partie inconnue : au delà des *connaissances du domaine* éventuelles, la structure se révèle par la relation entre *énoncés* ou *hypothèses* et représentations *extensionnelles*, c'est à dire les ensembles d'instances (données) reconnues par un énoncé du langage. Cette utilisation de la structure *via* la relation entre intensionnel et extensionnel se retrouvera dans plusieurs travaux présentés dans ce mémoire, En particulier nous verrons comment l'utilisation de

1. On préférera ici *instance* à *individu*

la *structure* cachée présente dans les données, et plus particulièrement dans les exemples positifs, permet de réduire l'espace de recherche et l'ensemble de solutions dans un problème d'apprentissage supervisé.

La notion d'*abstraction* est directement présente dans un ensemble de travaux où l'ensemble des représentations intensions/extensions est partiellement ordonné, les énoncés les plus généraux étant associés aux plus grandes *extensions*. Le niveau de granularité (grossière *vs* fine) des langages d'énoncés, ou des représentations extensionnelles, permet d'abstraire ou de raffiner ces ensembles de représentations. Il sera surtout question d'*abstraction extensionnelle*, et de ses conséquences sur la notion d'*implication*, c'est-à-dire indirectement sur la structure, dans le sens donné ci-dessus. Cette forme d'abstraction est moins pratiquée en Intelligence Artificielle que l'*abstraction intensionnelle* qui représente le passage d'un langage d'hypothèse à un langage moins détaillé. Elle consiste essentiellement à remplacer dans la notion d'extension, les objets par des groupes d'objets, ce qui est équivalent, formellement, à considérer des objets virtuels possédant chacun les caractéristiques communes à un groupe.

Enfin, on identifie en Apprentissage Artificiel un certain nombre de *paradigmes*, mot que nous emploierons ici dans le sens restreint² de *catégories de problèmes*, ou des manières de les aborder. On trouvera dans ce mémoire deux ensembles de travaux que l'on peut interpréter en ce sens, l'un traitant de la notion d'incomplétude ou d'ambiguïté des exemples, l'autre étendant l'apprentissage supervisé cohérent dans un cadre collectif. Dans les deux cas, de nouvelles notions doivent être introduites pour rendre compte des relations entre les observations et les hypothèses que l'on peut en induire.

Nous structurons le mémoire de la manière suivante : Le second chapitre porte sur les relations entre *intension* et *extension*, nous y rappellerons le cadre des correspondances de Galois, et les notions de *fermetures* associés, et traiterons de l'utilisation de la structure cachée pour réduire l'espace de recherche en apprentissage supervisé. Le troisième chapitre traitera dans ce cadre des notions d'abstraction extensionnelles et intensionnelles, et de leurs propriétés. Le quatrième est consacré à la recherche de motifs répétés, problème dans lequel la notion de motif a été progressivement étendue, et qui trouve ses motivations et ses applications en BioInformatique. Le cinquième traite des nouveaux paradigmes évoqués ci-dessus, partant d'un problème de BioInformatique dont la formulation fait apparaître la notion d'ambiguïté des objets, et finissant par l'exploration de l'apprentissage collectif. Enfin, dans la conclusion je trace quelques perspectives de recherches.

2. "...dans son livre La structure des révolutions scientifiques, Kuhn définit un paradigme scientifique comme suit : * un ensemble d'observations et de faits avérés, * un ensemble de questions en relation avec le sujet qui se posent et doivent être résolues, * des indications méthodologiques (comment ces questions doivent être posées), * comment les résultats de la recherche scientifique doivent être interprétés." (extraits de *Wikipedia*), mais attention, plus loin on lit, aussi, "... « paradigme » peut être considéré comme un *buzzword*, c'est-à-dire un mot prestigieux visant à intimider l'interlocuteur. " .

Chapitre 2

Structure : Intensions, Extensions, et Apprentissage supervisé

2.1 Intensions, Extensions, Ordres partiels

2.1.1 Extensions, Ordres partiels

Une grande partie de l'Intelligence Artificielle (I.A.) repose sur les liens entre les *énoncés* d'un langage permettant de décrire ou modéliser le monde, et un ensemble de situations représentant des états du monde. C'est en tous cas le propre des approches logiques de l'I.A. Lorsqu'il s'agit de modéliser le monde, l'approche *inductive* étudie plus précisément les liens entre les énoncés du langage et un ensemble d'*observations* tirées de la réalité. Nous serons confrontés plus précisément dans les problèmes rencontrés dans ce mémoire, à l'exploration d'un *Espace de Recherche*, constitué de représentations dite *intensionnelles* formées dans un langage \mathcal{L} , parmi lesquelles nous sélectionnerons des *hypothèses* ou des *hypothèses partielles* pour un problème donné. Nous disposerons également d'un ensemble d'observations I , dont les éléments seront appelés des *instances* et dont les parties sont des représentations dites *extensionnelles*. Le lien se fait de la manière suivante : chaque énoncé est satisfait par un ensemble d'instances, qui constitue son extension :

Définition 1 $ext_I(t) = \{x \in I \mid x \text{ satisfait } t\}$.

Nous supposons par la suite que la notion de *satisfaction d'un énoncé* t , et donc la fonction d'extension, est connue. Nous supposons également que tout élément de I appartient à l'ensemble (\mathcal{X}) représentant l'*Univers des Instances* et qui répond aux lois du monde observé, qu'on les connaisse, sous la forme d'une *théorie du domaine*, ou non.

Enfin, nous supposons ici que le langage \mathcal{L} est partiellement ordonné. On notera $t1 \succeq t2$ le fait que $t1$ soit *plus spécifique* que $t2$ dans le langage \mathcal{L} . On dira aussi dans ce cas que $t2$ est plus général que $t1$. Cette relation d'ordre sur le langage est dite *syntactique* ou *intensionnelle*.

Ce qui nous intéresse réellement en apprentissage de concepts est la relation d'ordre *sémantique* : c'est elle qui permet de définir l'*extension* du concept. Cependant lorsque nous parcourons l'ensemble des hypothèses qui est inclus dans \mathcal{L} , c'est à l'ordre sur \mathcal{L} que nous avons accès et ce

sont des éléments de \mathcal{L} que nous sélectionnerons comme résultat de l'apprentissage. Nous aurons toujours la propriété suivante :

Propriété 1

$$t1 \succeq t2 \Rightarrow \text{ext}_{\mathcal{X}}(t1) \subseteq \text{ext}_{\mathcal{X}}(t2)$$

Mais la réciproque n'est pas toujours vraie.

Si on représente par \mathcal{I} l'ensemble des objets que le langage \mathcal{L} peut distinguer en respectant la propriété 1¹, nous aurons :

$$t1 \succeq t2 \Leftrightarrow \text{ext}_{\mathcal{I}}(t1) \subseteq \text{ext}_{\mathcal{I}}(t2)$$

Cependant, certains objets que peut distinguer le langage ne peuvent pas être des instances de l'univers \mathcal{X} . Par exemple, un objet *blanc* \wedge *sombre* est une chimère : aucune instance de l'univers \mathcal{X} ne correspond à un tel objet.

Nous appellerons *Théorie du domaine* T l'ensemble de ce que nous savons de notre univers \mathcal{X} . \mathcal{X} est défini ici comme un sous-ensemble de \mathcal{I} . Nous supposons T représentable dans \mathcal{L} . Nous aurons alors en général, pour tout énoncé t de \mathcal{L} :

$$\text{ext}_{\mathcal{X}}(t) \subseteq \text{ext}_{\mathcal{I}}(t) \cap \text{ext}_{\mathcal{I}}(T) = \text{ext}_T(t)^2$$

Lorsque T n'est qu'une partie de ce que nous devrions savoir pour connaître \mathcal{X} , nous avons $\mathcal{X} \subsetneq \text{ext}_{\mathcal{I}}(T)$. Lorsque nous savons tout ce qui permet de déterminer les objets de \mathcal{I} qui sont des instances de \mathcal{X} , nous avons $\mathcal{X} = \text{ext}_{\mathcal{I}}(T)$. Dans ce dernier cas nous dirons que la théorie du domaine T est *complète*.

De plus un ordre extensionnel induit une relation d'équivalence sur \mathcal{L} :

$$t_1 \equiv t_2 \text{ ssi } \text{ext}(t_1) = \text{ext}(t_2)$$

.

Cette relation d'équivalence est dite *plus grossière* que l'égalité des énoncés au sens où :

$$t_1 = t_2 \Rightarrow t_1 \equiv t_2$$

Remarquons également qu'en apprentissage on n'a accès qu'à un sous-ensemble I des instances de \mathcal{X} : les *données*³. De ce fait on ne pourra différencier des hypothèses que si elles ont des extensions différentes dans ce sous-ensemble. Il est alors intéressant de considérer l'ordre extensionnel en se restreignant à I . Remarquons que lorsque les données augmentent, passant ainsi de I à $I' \supseteq I$, on se rapproche de l'ordre sur l'univers des Instances. On a ainsi :

$$t1 \succeq t2 \Rightarrow \text{ext}_{\mathcal{X}}(t1) \subseteq \text{ext}_{\mathcal{X}}(t2) \Rightarrow \text{ext}_{I'}(t1) \subseteq \text{ext}_{I'}(t2) \Rightarrow \text{ext}_I(t1) \subseteq \text{ext}_I(t2)$$

1. On considère ici que deux objets se distinguent si les assignations de *Vrai* (l'objet satisfait l'énoncé) et *Faux* à tous les éléments de \mathcal{L} sont différentes.

2. Par abus de langage on écrit donc $\text{ext}_T(t)$ en assimilant T à l'ensemble $\text{ext}_{\mathcal{I}}(T)$.

3. il s'agit ici d'un abus de langage : un ensemble de données I peut contenir plusieurs observations de même description dans \mathcal{X} , c'est donc l'ensemble de ces descriptions qui est inclus dans \mathcal{X} .

On associe ainsi à I la relation d'équivalence \equiv_I , telle que :

$$t_1 \equiv_I t_2 \Leftrightarrow \text{ext}_I(t_1) = \text{ext}_I(t_2)$$

et on remarque que ces relations d'équivalences sont ordonnées ainsi de la plus *fine* à la plus *grossière* :

$$t_1 = t_2 \Rightarrow t_1 \equiv_{\mathcal{X}} t_2 \Rightarrow t_1 \equiv_I t_2$$

Ce qui signifie que des énoncés de \mathcal{L} qui sont distincts sur l'univers des instances se confondent extensionnellement sur les données I dont on dispose. Remarquons aussi que $\text{ext}_I(t_1) \subseteq \text{ext}_I(t_2)$ signifie que l'implication $t_1 \rightarrow t_2$ est valide sur I , autrement dit dans une classe d'équivalence de \equiv_I tous les termes s'impliquent deux à deux, et notamment les termes *minimaux* impliquent les termes *maximaux*. Il y a donc des implications qui ne sont ni liées au langage, ni liées à l'ajout d'une théorie du domaine au langage, mais *contingentes* aux données. Certaines sont parfois des artefacts : elles sont valides sur I mais pas sur $I' \supseteq I$ obtenu en ajoutant des données à I . Ainsi, supposons qu'une classe de \equiv_I est formée de $\{\text{oiseau}, \text{oiseau-bleu}\}$, ce qui signifie que $\text{oiseau} \rightarrow \text{oiseau-bleu}$ est valide dans I , mais on peut s'apercevoir que $\text{oiseau} \rightarrow \text{oiseau-bleu}$ n'est plus valide lorsqu'on ajoute un oiseau (qui se trouve être blanc) à I .

2.1.2 Intentions et Extensions

Il est alors intéressant de se restreindre, lorsqu'on explore \mathcal{L} , à des représentants des classes d'équivalences de \equiv_I . Nous allons d'abord supposer qu'il existe un *unique élément maximal* dans \mathcal{L} parmi tous ceux dont l'extension contient un sous-ensemble donné e de I . Nous noterons cet élément $\text{int}(e)$, et l'appellerons l'*intension* de e .

On obtient cette propriété naturellement lorsque deux conditions sont remplies :

1. On peut associer à toute instance x de I un *unique terme maximale* spécifiquement, que nous noterons $\dot{x} = \text{int}(\{x\})$, parmi tous les termes t tels que $x \in \text{ext}_I(t)$, ce qui amène à $x \in \text{ext}_I(t) \Leftrightarrow \dot{x} \succeq t$.
2. \mathcal{L} est un inf-demi-treillis (voir Définition 2) : toute paire t_1, t_2 d'éléments de \mathcal{L} a un unique plus grand minorant, noté $t_1 \wedge_{\mathcal{L}} t_2$: on a $t_1 \succeq t_1 \wedge_{\mathcal{L}} t_2$, $t_2 \succeq t_1 \wedge_{\mathcal{L}} t_2$ et pour tout l , $t_1 \succeq l$ et $t_2 \succeq l$ implique $t_1 \wedge_{\mathcal{L}} t_2 \succeq l$, ce qui suppose un unique élément minimal, noté \perp . Ce plus grand minorant est aussi appelé, suivant les travaux, *lcs* (*least common subsumer*), ou *msg* (*maximally specific generalization*), ou encore *lgg* (*least general generalisation*).

La première condition, connue sous le nom de *single trick representation*, permet en particulier de remplacer en apprentissage le test $x \in \text{ext}(t)$ par $t \preceq \dot{x}$, c'est à dire t est plus général que \dot{x} , ou encore t *subsume* \dot{x} . La seconde permet d'obtenir $\text{int}(e)$:

Proposition 1 Soient une fonction int , un ensemble I et un langage \mathcal{L} satisfaisant les conditions 1 et 2 ci-dessus. Pour toute partie $e = \{x_1, \dots, x_n\}$ de I , si int est définie pour toute partie $e = \{x_1, \dots, x_n\} \subseteq I$ par :

$$\text{int}(e) = \text{int}(\{x_1\}) \wedge_{\mathcal{L}} \dots \wedge_{\mathcal{L}} \text{int}(\{x_n\})$$

alors $\text{int}(e)$ est le plus spécifique parmi les termes t de \mathcal{L} tels que $e \subseteq \text{ext}(t)$.

Preuve

Posons $int(e) = int(\{x_1\}) \wedge_{\mathcal{L}} \cdots \wedge_{\mathcal{L}} int(\{x_n\})$. Par définition $int(e)$ est le plus grand minorant des $int(\{x_i\})$, donc :

- $\forall t$ tel que $\forall x_i \ int(\{x_i\}) \succeq t$, on a $int(e) \succeq t$. Or par définition, $int(\{x\}) \succeq t \Leftrightarrow x \in ext(t)$, donc $\forall t$ tel que $\forall x_i, \ x_i \in ext(t)$, on a $int(e) \succeq t$.
- $\forall x_i, x_i \in ext(int(e))$.

On en conclut que $int(e)$ est le plus grand élément de \mathcal{L} dont l'extension contient $e = \{x_1, \dots, x_n\}$. \square

Nous allons maintenant voir que $\mathcal{P}(I)$, \mathcal{L} , int et ext sont liés dans une *correspondance de Galois*. Les principales définitions, preuves et résultats concernant les correspondances et treillis de Galois sont présentées entre autres dans [Bir73],[DP02], et, reflétant l'activité pionnière de l'école française de Mathématiques pour les Sciences sociales, dans [BM70, Mon03], En particulier le livre récent et très complet de N. Caspard, B. Leclerc et B. Monjardet [CLM07] ainsi que l'article de revue de N. Caspard et B. Monjardet [CM03] sont d'une lecture très enrichissante, de même que le résumé des travaux de V. Duquenne [Duq99], notamment sur la base d'implications minimale associée à un treillis de Galois. D'autres résultats et algorithmes sur les treillis de Galois redéfinis dans le champ, très actif également, de l'Analyse Formelle de Concepts apparaissent dans [GW99]. Dans ces dernières présentations, le treillis de Galois est défini relativement à la correspondance de Galois d'une relation entre deux ensembles I et A , où A est un ensemble d'*attributs*, et est donc défini entre $\mathcal{P}(I)$ et $\mathcal{L} = \mathcal{P}(A)$, avec l'inclusion ensembliste comme ordre partiel. Cependant, dans le domaine de l'Apprentissage Artificiel, J-G. Ganascia a proposé une formulation de l'apprentissage supervisé et de la construction de théories, utilisant une correspondance de Galois entre un langage \mathcal{L} plus général qu'un ensemble de parties, et $\mathcal{P}(I)$ [Gan93]. M. Liquière et J. Sallantin ont étendu cette formulation à des langages dont les éléments sont des graphes, et proposé un algorithme de construction du treillis de Galois correspondant [LS98]. Plus récemment, l'Analyse des Objets Symboliques[BD00], et l'Analyse des Concepts Logiques de S. Ferré [CFRD07, CFRD08] définissent le treillis de Galois de correspondances portant également, pour la partie intensionnelle, sur des langages plus larges. Nous avons proposé une présentation étendue de la notion de treillis de Galois [PRSV02], dans le but de faire varier, d'une part, la notion d'*extension*, en restreignant ainsi $\mathcal{P}(I)$ à un sous-ensemble $E \subseteq \mathcal{P}(I)$, de manière à introduire une notion d'*abstraction extensionnelle*, et d'autre part, le langage \mathcal{L} utilisé et qui pourra également être soumis à une forme d'*abstraction intensionnelle*. Dans la suite du papier nous appelons treillis de Galois la structure formelle que nous définissons ci-dessous et qui est simplement le treillis de la correspondance de Galois entre deux treillis quelconques, et réservons le terme *treillis de concepts* aux treillis de Galois présentés dans [GW99].

2.1.3 Correspondance de Galois

Définition 2 (Treillis) *Un ensemble partiellement ordonné (P, \leq) est*

- *un sup-demi-treillis si à toute paire d'éléments (x, y) de P on peut associer une borne supérieure (ou unique plus petit majorant) $x \vee y$:*
 $x \vee y \geq x, \ x \vee y \geq y$ *et pour tout u tel que $u \geq x$ et $u \geq y$ on a $u \geq x \vee y$.*

- un *inf-demi-treillis* si à toute paire d'éléments (x, y) de P on peut associer une borne inférieure (ou unique plus grand minorant) $x \wedge y$:
 $x \wedge y \leq x$, $x \wedge y \leq y$ et pour tout u tel que $u \leq x$ et $u \leq y$ on a $u \leq x \wedge y$.
- un *treillis* s'il est à la fois un *sup-demi-treillis* et un *inf-demi-treillis*

On remarque que pour un obtenir un treillis à partir d'un inf-demi-treillis fini P , il suffit de s'assurer qu'il y a un unique élément maximal dans P , ou de l'ajouter. Nous noterons \top cet élément maximal. En effet, dans ce cas il y a également un unique plus petit majorant $t_1 \vee t_2$ pour toute paire d'éléments $\{t_1, t_2\}$. La figure 2.1 en donne la preuve.

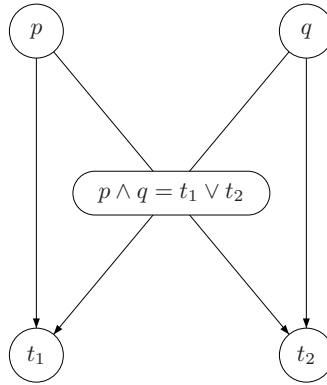


FIGURE 2.1 – Si p et q sont deux majorants minimaux de t_1 et de t_2 , alors $p \wedge q$ est aussi majorant de t_1 et de t_2 : en effet $p \wedge q$ est la borne inférieure (plus grand minorant) de p et de q , et t_1 et t_2 étant aussi des minorants de p et de q , nécessairement $p \wedge q$ est plus grand que t_1 et plus grand que t_2 . On en déduit que $p \wedge q$ étant plus petit que p et q , qui étaient tous les deux minimaux, on a donc $p = q = t_1 \vee t_2$. Notons qu'un majorant de t_1 et de t_2 existe toujours puisque \top est un majorant de tous les éléments de P .

Le même raisonnement permet d'obtenir un treillis à partir d'un sup-demi-treillis fini. Nous utiliserons cette propriété pour construire des *abstractions* au chapitre suivant :

Proposition 2 Soit P un ensemble fini partiellement ordonné pour la relation \leq_P .

- Si P est un *inf-demi-treillis*, alors $P \cup \{\top\}$, où \top est maximal pour \leq_P dans P , est un *treillis*.
- Si P est un *sup-demi-treillis*, alors $P \cup \{\perp\}$, où \perp est minimal pour \leq_P dans P , est un *treillis*.

Nous aurons besoin dans ce qui suit pour le langage intensionnel \mathcal{L} , et au chapitre 3 pour $\mathcal{P}(I)$ de la définition des éléments \vee -irréductibles ("ou" irréductibles, ou encore sup-irréductibles) d'un treillis. Les éléments \vee -irréductibles d'un sup-demi-treillis T sont ceux, excepté \perp qu'on ne peut engendrer en appliquant l'opérateur \vee à d'autres éléments de T ⁴. [DP02, BM70]. Il est d'usage de les définir pour un treillis :

Définition 3 Soit (T, \leq) un treillis fini, un élément t de T est dit \vee -irréductible si :

4. Les éléments \wedge -irréductibles se définissent d'une manière duale.

- $t \neq \perp$ et
- $t = a \vee b$ implique $t = a$ ou $t = b$.

Remarquons qu'un sup-demi-treillis étant par définition clos par l'opérateur \vee , les éléments \vee -irréductibles permettent d'engendrer tout le sup-demi-treillis (à l'exception de l'élément minimal \perp , si celui-ci existe). Par exemple, dans le treillis $\mathcal{P}(I)$ des parties d'un ensemble I fini, les éléments \vee -irréductibles sont les éléments de I : toute partie est obtenue comme réunion de ces éléments⁵. Dans ce cas il s'agit des éléments minimaux mais ce n'est pas toujours vrai⁶.

Définition 4 (Correspondance de Galois) Soient $m1 : P \rightarrow Q$ et $m2 : Q \rightarrow P$ des fonctions définies sur deux ensembles ordonnés (P, \leq_P) et (Q, \leq_Q) . $(m1, m2)$ est une correspondance de Galois si pour tout $p, p1, p2$ de P et pour tout $q, q1, q2$ de Q :

- C1- $p1 \leq_P p2 \Rightarrow m1(p2) \leq_Q m1(p1)$
- C2- $q1 \leq_Q q2 \Rightarrow m2(q2) \leq_P m2(q1)$
- C3- $p \leq_P m2(m1(p))$ et $q \leq_Q m1(m2(q))$

Nous rappelons ci-dessous la définition d'un opérateur de fermeture :

Définition 5 (Fermeture) w est un opérateur de fermeture sur un ensemble partiellement ordonné (P, \leq) ssi pour tout couple (x, y) d'éléments de P on a :

- $x \leq w(x)$ (extensivité)
- Si $x \leq y$ alors $w(x) \leq w(y)$ (monotonie)
- $w(x) = w(w(x))$ (idempotence)

Un élément de P tel que $x = w(x)$ est appelé un élément fermé (ou clos) relativement à w .

Le treillis de Galois $G(P, m1, Q, m2)$ dont chaque élément est un couple de fermés respectivement de P et de Q est défini de la manière suivante :

Proposition 3 (Treillis de Galois) Soient $m1 : P \rightarrow Q$ et $m2 : Q \rightarrow P$ deux fonctions définies sur les treillis (P, \leq_P) et (Q, \leq_Q) , tels que $(m1, m2)$ est une correspondance de Galois.

Soit $G = \{(p, q)\}$, où p est un élément de P et q un élément de Q , tels que $p = m2(q)$ et $q = m1(p)$

Soit \leq la relation définie par : $(p1, q1) \leq (p2, q2)$ ssi $q1 \leq_Q q2$.

(G, \leq) est un treillis appelé treillis de Galois avec :

- $(p1 \wedge_P p2, m1 \circ m2(q1 \vee_Q q2))$ comme borne supérieure et
- $(m2 \circ m1(p1 \vee_P p2), q1 \wedge_Q q2)$ comme borne inférieure

Nous avons alors les propriétés suivantes :

Proposition 4

1. Dans une correspondance de Galois, $m2 \circ m1$ et $m1 \circ m2$ sont des opérateurs de fermeture respectivement pour P et Q .
2. Soient $p \in P$ et $q \in Q$ alors $m2(q)$ et $m1(p)$ sont des éléments fermés respectivement de P et de Q .

5. Précisément par réunion des singletons de $P(I)$.

6. Par exemple, dans un treillis d'intervalles, les éléments \vee -irréductibles ne sont pas les éléments minimaux.

3. On a $m_1(m_2 \circ m_1(p)) = m_1(p)$ et $m_2(m_1 \circ m_2(q)) = m_2(q)$.
4. Si on considère les relations d'équivalences \equiv_P et \equiv_Q définies respectivement par $p \equiv_{m_1} p'$ ssi $m_1(p) = m_1(p')$ et $q \equiv_{m_2} q'$ ssi $m_2(q) = m_2(q')$, alors $m_2(m_1(p))$ et $m_1(m_2(q))$ sont respectivement l'unique plus grand élément de la classe d'équivalence de \equiv_{m_1} qui contient p et l'unique plus grand élément de la classe d'équivalence de \equiv_{m_2} qui contient q .
5. Pour toute paire p_1, p_2 , de P on a $m_1(p_1) \wedge_Q m_1(p_2) = m_1(p_1 \vee_P p_2)$
6. Pour toute paire q_1, q_2 , de Q on a $m_2(q_1) \wedge_P m_2(q_2) = m_2(q_1 \vee_Q q_2)$

2.1.4 Treillis de Galois des intensions/extensions sur $\mathcal{P}(I)$

Considérons le langage $P = \mathcal{L}$, l'ensemble $Q = \mathcal{P}(I)$ des parties de l'ensemble d'instances I . ainsi que les fonctions $m_1 = \text{ext}$ et $m_2 = \text{int}$ décrites au paragraphe 2.1.2. \mathcal{L} étant un inf-demi-treillis est aussi un treillis d'après la proposition 2 en lui ajoutant si nécessaire un élément maximal \top . $\mathcal{P}(I)$ est un treillis pour \subseteq avec $\vee_Q = \cup$ et $\wedge_Q = \cap$.

On a alors :

- C1 : $t_1 \preceq t_2 \Rightarrow \text{ext}(t_2) \subseteq \text{ext}(t_1)$ d'après la proposition 1
- C2 : $e_1 \subseteq e_2 \Rightarrow \text{int}(e_2) \preceq \text{int}(e_1)$. En effet, on a alors $e_2 = e_1 \cup (e_2 - e_1)$ et donc, $\text{int}(e_2) = \text{int}(e_1) \wedge_{\mathcal{L}} \text{int}(e_2 - e_1)$ d'après la proposition 1 et donc $\text{int}(e_2) \preceq \text{int}(e_1)$ d'après la définition de $\wedge_{\mathcal{L}}$.
- C3-
 - $t \preceq \text{int}(\text{ext}(t))$
 Soit $\text{ext}(t) = \{x_1, \dots, x_n\}$. Selon la proposition 1, on a $\text{int}(\text{ext}(t)) = \text{int}(\{x_1\} \wedge_{\mathcal{L}} \dots \wedge_{\mathcal{L}} \text{int}(\{x_n\}))$. Or selon la condition 1 caractéristique de int (cf proposition 1), on a pour tout i , $\text{int}(\{x_i\}) \succeq t$. On en déduit que $\text{int}(\text{ext}(t)) \succeq t$.
 - $e \subseteq \text{ext}(\text{int}(e))$
 A tout $x \in e$ on associe, d'après la condition 1, $\text{int}(\{x\})$ l'unique plus grand élément de \mathcal{L} dont l'extension contient x . On a donc $\{x\} \subseteq \text{ext}(\text{int}(\{x\}))$. Par ailleurs comme $\{x\} \subseteq e$, on a d'après C2 $\text{int}(e) \preceq \text{int}(\{x\})$ et donc, d'après C1, $\text{ext}(\text{int}(\{x\})) \subseteq \text{ext}(\text{int}(e))$. On en conclut que pour tout x de e , $\{x\} \subseteq \text{ext}(\text{int}(e))$, et donc $e \subseteq \text{ext}(\text{int}(e))$.

On en conclut :

Proposition 5 (ext, int) forme une correspondance de Galois sur les treillis \mathcal{L} et $\mathcal{P}(I)$

En découlent les propriétés suivantes pour deux fermés t_1 et t_2 de \mathcal{L} :

1. Leur borne supérieure dans le treillis des fermés de \mathcal{L} , c'est à dire le moindre généralisé de t_1 et de t_2 , est $(t_1 \wedge_{\mathcal{L}} t_2)$.
2. Leur borne inférieure dans le treillis des fermés de \mathcal{L} , c'est à dire le moindre spécialisé de t_1 et de t_2 , est $\text{int} \circ \text{ext}(t_1 \vee_{\mathcal{L}} t_2)$.
3. $\text{int}(e_1 \cup e_2) = \text{int}(e_1) \wedge_{\mathcal{L}} \text{int}(e_2)$.
4. $\text{ext}(t_1 \vee_{\mathcal{L}} t_2) = \text{ext}(t_1) \cap \text{ext}(t_2)$.

Les propriétés 1 et 3 permettront une exploration ascendante (bottom-up, du plus spécifique au plus général) d'un espace de termes fermés, en partant des exemples positifs. La propriété

2 permet l'exploration descendante (du plus général au plus spécifique) de ce même espace. La propriété 4 permet de parcourir un espace d'hypothèses en faisant des intersections d'extensions, ce qui sera utilisé au chapitre 4 pour la recherche de motifs répétés.

Le treillis de Galois $G(\mathcal{L}, ext, \mathcal{P}(I), int)$ servira par la suite d'origine à un ensemble partiellement ordonné de treillis *abstrait*, intensionnellement ou extensionnellement. La proposition 4 dit en particulier que la classe d'équivalence de $\equiv_{\mathcal{L}}$ correspondant à l'extension $e \subseteq I$, a comme représentant un énoncé fermé c pour $int \circ ext$: l'élément le plus spécifique de la classe, et appartient en conséquence au noeud (e, c) du treillis de Galois $G(\mathcal{L}, ext, \mathcal{P}(I), int)$.

2.2 Exploration d'un espace quotient en apprentissage supervisé

Nous nous sommes intéressés à l'espace de recherche parcouru, dans le cadre de l'apprentissage supervisé de type *apprentissage par règles*. Selon la stratégie *separate and conquer* [Für99] à chaque itération l'espace d'hypothèses *partielles*, que nous appellerons par la suite *termes*, est exploré à la recherche d'un *meilleur* terme conjonctif (ou d'une meilleure *règle*) selon une certaine fonction d'évaluation. De nombreux systèmes d'apprentissage par règles sont fondés sur l'exploration de cet espace, soit de manière *descendante* (*top-down*) (par exemple CN2[CB91], RIPPER[Coh95], AQ[WMKP06]), c'est à dire en partant de l'énoncé le plus général ($\perp_{\mathcal{L}}$), soit de manière *ascendante* (*bottom-up*) (DLG[WA92], Golem[MF90], LCSLearnDisj[CH94]) ou bidirectionnelle (JOJO [FW93]). Quelques systèmes, comme RISE[Dom96], utilisent une stratégie globale : une hypothèse complète, constituée de plusieurs termes conjonctifs, est recherchée.

2.2.1 Langage et Apprentissage

Nous traitons dans ce qui suit essentiellement de langages \mathcal{L} propositionnels ou d'ordre 0, par opposition aux langages relationnels ou d'ordre 1. En apprentissage supervisé *symbolique*, ce qui signifie ici que l'on cherche les solutions dans un langage d'hypothèses \mathcal{L} soumis à un ordre partiel, les exemples sont en général décrits sous la forme d'une table *attribut-valeur* dont les attributs sont de différents types en fonction du domaine des valeurs : booléennes, nominales, ordonnées ou numériques. Le langage \mathcal{L} dans lequel on cherche les solutions est en partie contraint par cette représentation, mais résulte d'un choix, éclairé par les connaissances concernant le problème⁷.

Remarquons que la convention veut qu'on considère en apprentissage l'ordre partiel du plus général (élément d'extension maximale, *sommet* ou *top*) au plus spécifique (élément minimal, *base* ou *bottom*), c'est à dire suivant l'ordre extensionnel. Notre élément minimal $\perp_{\mathcal{L}}$ devient alors l'élément maximal selon cet ordre, d'où le terme *top-down*. Cette convention va dans le sens de la dénomination des opérateurs *ET* et *OU* en logique qui suivent aussi naturellement cet ordre extensionnel⁸. Ainsi, dans un langage de termes conjonctifs, $(a \text{ ET } b) \vee_{\mathcal{L}} (b \text{ ET } c)$ correspond à la moindre spécialisation de $(a \text{ ET } b)$ et de $(b \text{ ET } c)$, c'est dire $(a \text{ ET } b \text{ ET } c)$. De même, $(a \text{ ET } b) \wedge_{\mathcal{L}} (b \text{ ET } c)$ est la généralisation maximalement spécifique de $(a \text{ ET } b)$ et de $(b \text{ ET } c)$,

7. Nous verrons au chapitre suivant que l'on peut choisir celui-ci dans un espace de langages $\{\mathcal{L}_i\}$ lui même partiellement ordonné.

8. Les animaux grands ET stupides sont l'intersection des animaux grands ET des animaux stupides.

c'est à dire ici b . Dans la suite de ce chapitre, nous noterons ET , OU les opérateurs logiques.

Une manière générique de procéder consiste à associer chaque attribut a à un treillis de parties T_a de son domaine D_a . Ainsi si a est une forme avec $D_a = \{rectangle, losange, triangle\}$, on peut considérer $T_a = \{rectangle, losange, triangle, quadrilatère, \top_a, \perp_a\}$ où $quadrilatère$ représente $\{rectangle, losange\}$, et \top et \perp respectivement D_a et \emptyset . L'attribut a est ici représenté par une hiérarchie. Un exemple de terme de \mathcal{L} serait par exemple $t = forme-quadrilatère$. Supposons également que b soit une couleur parmi $\{bleu, rouge, vert\}$ et qu'on considère T_b constitué seulement de ces valeurs et de D_b et \emptyset . Un autre exemple de terme de \mathcal{L} est alors $t = forme-quadrilatère ET-couleur-rouge$. Ici on a choisi de ne pas considérer les sous-ensembles de 2 couleurs dans \mathcal{L} .

Les éléments $\vee_{\mathcal{L}}$ -irréductibles sont alors ceux qui permettent de spécialiser un terme courant. Il y a une unique représentation maximale de tout élément de \mathcal{L} en éléments $\vee_{\mathcal{L}}$ -irréductibles, et lorsque le treillis est distributif⁹, il y a également une unique représentation minimale. Les spécialisations envisagées sont alors de la forme $(t \vee_{\mathcal{L}} o)$ où o est un élément $\vee_{\mathcal{L}}$ -irréductible, ce qui assure de considérer tout \mathcal{L} lors d'une exploration par spécialisation. Pour reprendre l'exemple précédent, à chaque attribut est associé un ensemble d'éléments $\vee_{\mathcal{L}}$ -irréductibles, et pour l'attribut *forme*, il s'agit d'une hiérarchie et donc tous les éléments sont $\vee_{\mathcal{L}}$ -irréductibles (ne peuvent être obtenus par spécialisation) sauf $\perp_{\mathcal{L}}$ (qui est exclus par définition) et $\top_{\mathcal{L}}$ (qui comme élément maximal est obtenu comme $\vee_{\mathcal{L}}$ de toutes les feuilles). Il s'agit donc de $\mathcal{O} = \{forme-rectangle, forme-losange, forme-triangle, forme-quadrilatère\}$. La figure Figure 2.2 représente ce treillis de formes ainsi qu'un treillis d'intervalles.

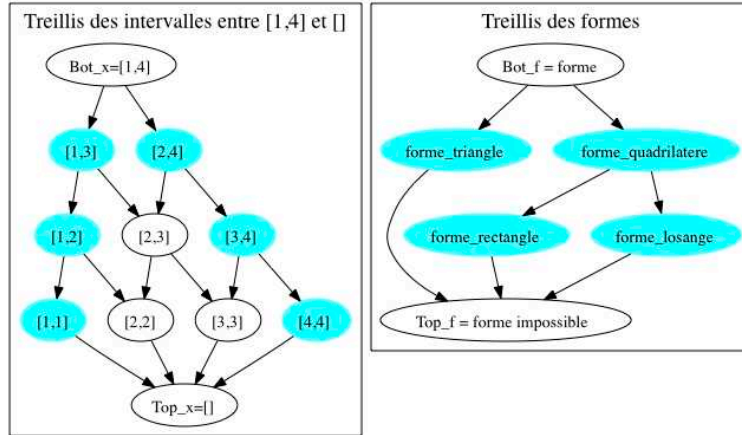


FIGURE 2.2 – A gauche un treillis d'intervalles représentant le domaine d'un attribut numérique, à droite une hiérarchie. En bleu (ou gris) les noeuds $\vee_{\mathcal{L}}$ -irréductibles. Dans le cas numérique, par exemple, $x \in [2, 3]$ n'est pas irréductible car il peut être obtenu comme $x \in [1, 3] \vee_{\mathcal{L}} x \in [2, 4]$.

De cette manière les connaissances du domaine sur les attributs sont intégrées à la structure de \mathcal{L} [Gan93]. Cependant les programmes d'apprentissage de référence ne permettent pas une telle souplesse de représentation, et se limitent à associer à chaque attribut le type nominal ou

9. c'est-à-dire pour tout triplet d'éléments de \mathcal{L} , a, b, c on a $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$, ce qui est le cas par exemple pour les treillis isomorphes au treillis $\mathcal{P}(A)$ des parties d'un ensemble A , ou pour le treillis des intervalles entiers entre un singleton $[i, i]$ et $[1, \dots, i, \dots, n]$, ainsi que pour les produits cartésiens de tels treillis

numérique. Une alternative, que nous utilisons dans les programmes présentés au paragraphe suivant, est d'associer à chaque élément du treillis de valeur de chaque attribut un atome, représentant la contrainte correspondante. On obtient ainsi un treillis atomique (en bijection avec le treillis des parties de l'ensemble des atomes). Dans l'exemple ci-dessus un exemple *rectangle rouge* x sera représenté par $\dot{x} = \text{rectangle } ET \text{ quadrilatère } ETrouge$. Dans le cas de l'apprentissage d'une fonction booléenne nous aurons deux atomes $x = V$, $x = F$, ou plus simplement x et $not - x$, pour chaque attribut. La représentation du treillis d'intervalles d'un attribut numérique x est faite par un ensemble d'atomes de la forme $\{x \leq x_i, x > x_i\}$. La théorie du domaine, qui inclut ici des contraintes cachées comme $(x \leq x_i \text{ } ET \text{ } x_i < x_k \Rightarrow x \leq x_k)$ n'est plus intégrée à \mathcal{L} . Cependant elle est implicite dans la représentation des exemples, et donc dans le treillis de Galois associé aux exemples.

2.2.2 Apprentissage dans un espace quotient

Nous avons mené avec Pierre Brézellec et Marc Champesme une série de travaux sur l'apprentissage par exploration d'un espace de recherche organisé par un ordre partiel tel que défini au paragraphe 2.1.1. Dans ce qui suit, nous considérons que nous avons à faire à un problème d'apprentissage d'un concept représentable dans un langage $H = \mathcal{L}^m$, où $m \leq |\mathcal{X}|$: un concept devra donc être représentable par un énoncé h d'au plus m termes h_i de \mathcal{L} . On dispose d'un ensemble d'exemples E munis d'un label $+$ ou $-$ permettant de partitionner E en E^+ et E^- . Les exemples de E^+ devront être dans l'extension de l'hypothèse solution $h \in H$, et ceux de E^- ne pas être dans l'extension de h . Nous écrirons $h = h_1 \text{ } OU \text{ } \dots \text{ } OU \text{ } h_n$ avec la sémantique classique :

$$x \in ext(h_1 \text{ } OU \text{ } h_2) \text{ ssi } x \in ext(h_1) \text{ ou } x \in ext(h_2)$$

où chaque h_i est un terme du langage \mathcal{L} que nous supposons vérifier les conditions de la proposition 1 et donc vérifier la proposition 2 : (int, ext) forme une correspondance de Galois.

Toute exploration d'un tel espace de recherche H part d'un élément h de H et suit l'ordre partiel à l'aide d'*opérateurs de spécialisation* pour les approches top-down, ou d'*opérateurs de généralisation* pour les approches bottom-up.

Avec Pierre Brézellec, nous avons travaillé d'abord sur des algorithmes bottom-up. Dans ce type d'algorithme, l'opérateur élémentaire de généralisation dans \mathcal{L} est la *généralisation maximalement spécifique* (*msg* ou *lcs* ou encore *lgg*) dans \mathcal{L} . Précisément, on considère les généralisations possibles d'un terme courant t , par un exemple positif représenté par son plus petit généralisé \dot{x} . Par construction un tel terme s'écrit $t = int(e)$, où e est une partie de E^+ , et sa généralisation maximalement spécifique avec x s'écrit donc $t' = int(e \cup \{x\}) = t \wedge_{\mathcal{L}} \dot{x}$ ¹⁰. L'espace exploré est donc réduit, relativement à \mathcal{L} .

Dans notre premier travail, l'espace parcouru est celui des hypothèses complètes de H , $h = h_1 \text{ } OU \text{ } \dots \text{ } h_n$. Le système SAMIA [BS93b] utilise des opérateurs de généralisation et de spécialisations d'une hypothèse complète : des termes h_i peuvent être ôtés de h (généralisation), ou ajoutés (spécialisation), ou encore modifiés par généralisation par $t \wedge_{\mathcal{L}} \dot{x}$, l'ensemble des modifications possibles représentant le voisinage de h . Après l'application de chaque opérateur, on vérifie qu'un des termes n'est pas plus spécifique qu'un autre, auquel cas il est supprimé, réduisant ainsi la

10. c'est-à-dire $lgg(t, \dot{x})$.

taille de h . L'exploration utilise une méthode de *recuit simulé* [GKR94] : un voisin de l'hypothèse courante h est tiré au hasard, s'il est meilleur que h (au sens d'une fonction d'évaluation) il remplace celui-ci, sinon avec une probabilité décroissante au cours de l'exploration, il peut tout de même être choisi. La probabilité étant décroissante, l'exploration finit par trouver un minimum local de la fonction d'évaluation. Celle-ci est l'*erreur empirique* $err_{emp}(h)$, une estimation, sur les données d'apprentissage, de la probabilité d'erreur de classement par h d'une nouvelle instance. Nous utiliserons également par la suite comme fonction d'évaluation la *justesse empirique* $J_{emp}(p) = 1 - err_{emp}(h)$. Pendant son doctorat M'hammed Henniche a développé MG¹¹ une méthode globale bottom-up plus directe que SAMIA, proche de RISE et DLG, et utilisable dans une approche incrémentale. La méthode résultante, MGI [Hen94, Hen97], sera utilisée par la suite comme mécanisme de révision dans une méthode d'apprentissage collectif décrite au chapitre 5. La préoccupation dans cette méthode *en-ligne* est de minimiser les modifications à faire pour réviser une solution courante lorsque de nouveaux exemples sont reçus.

Cependant l'exploration globale est parfois difficile à optimiser, lorsque l'espace de recherche est grand. Nous avons alors utilisé dans ELENA [BS93a] la construction séquentielle des termes basée sur l'algorithme glouton de recouvrement de l'ensemble E^+ par un sous-ensemble minimal de parties. Cet algorithme doit son succès à ses garanties d'approximation très intéressantes pour un algorithme polynomial [Vaz06] et est au coeur de la stratégie *separate and conquer*, avec un certain nombre de variantes. Le principe est simple : on cherche un meilleur terme h_1 , au sens où il est cohérent et maximise le nombre d'exemples positifs *couverts* (c'est-à-dire qui satisfont h). Ces exemples couverts sont retirés de E^+ et on cherche un deuxième terme cohérent et maximisant le nombre d'exemples de E^+ couverts, et ainsi de suite jusqu'à ce que E^+ soit vide. On obtient donc, sauf si c'est impossible, une hypothèse $h = h_1 OU \dots OU h_n$ complète et cohérente et dont le nombre de termes est (approximativement) minimal. Dans ELENA, comme dans Golem ou LCSLLearnDisJ, chaque recherche d'un meilleur terme h_i se fait avec une exploration bottom-up de type *beam-search*. On part d'exemples positifs plongés dans \mathcal{L} , $\{\dot{x} \dots\}$, qui initialisent une liste de candidats à la généralisation, et de leurs voisins par généralisation minimale, $\{(\dot{x} \wedge \dot{x}'), \dots\}$, parmi lesquels on sélectionne les k meilleurs selon la fonction d'évaluation. Il s'agit ici de la *justesse empirique* à laquelle est ajoutée la *contrainte de cohérence* : aucun (ou éventuellement moins de p) exemple négatif ne doit satisfaire le candidat $\dot{x} \wedge \dot{x}'$. Les k énoncés retenus $\{h^1, \dots, h^k\}$ forment la nouvelle liste de candidats, et sont à leur tour minimalement généralisés en $\{(h^i \wedge \dot{x}') \dots\}$, termes parmi lesquels on sélectionnera de nouveau les k meilleurs candidats. On maintient au cours de l'exploration le meilleur candidat trouvé h_{sol} . Lorsque la liste de candidats devient vide, $h_i = h_{sol}$ est l'hypothèse partielle ajoutée à l'hypothèse courante h .

C'est également au cours de ce travail qu'est apparue clairement le fondement théorique de cette réduction de l'espace de recherche aux généralisations minimales de parties de E^+ . Les éléments de cet espace réduit sont appelés *prototypes* et ne sont autres que les représentants maximaux des classes d'équivalences de $\equiv ext_{E^+}$ (définies sur ext_{E^+}), et donc les fermés de \mathcal{L} relativement à $int \circ ext_{E^+}$, autrement dit les *intensions* des noeuds du treillis de Galois $G(\mathcal{L}, ext_{E^+}, \mathcal{P}(E^+), int)$. Cet espace réduit a la propriété suivante :

Proposition 6 *pour tout terme t de l'espace initial, il y a un prototype p tel que $ext_{E^+}(p) =$*

11. MG a eu un précurseur, APPTMS, dont les opérateurs étaient issus d'une inversion des opérateurs de l'ATMS de De Kleer.

$ext_{E^+}(t)$ et $ext_{E^-}(p) \subseteq ext_{E^-}(t)$, et donc $J_{emp}(p) \succeq J_{emp}(t)$.

Cela signifie que l'espace réduit contient donc forcément le terme maximisant J_{emp} .

Cependant, à l'encontre des solutions trouvées par les méthodes top-down, pour lesquelles on arrête la spécialisation dès que possible, un prototype p peut être équivalent, sur E , à un terme t plus général. Pour donner un exemple simple, supposons que nous ayons deux exemples positifs, dans un langage sans négation, $x_1 = abc$ et $x_2 = abd$ et un exemple négatif be . Le prototype trouvé sera $x_1 \wedge_{\mathcal{L}} x_2 = ab$ et il est tel que $ext_E(ab) = ext_E(a)$. Le terme a est un terme maximale-ment général de la classe d'équivalence de ab sur E :

Proposition 7 *Si t est un fermé de ext_{E^+} , t est aussi un fermé de ext_E .*

Soit une classe d'équivalence de \equiv_{ext_E} , d'extension $e \subseteq E$.

- *Le prototype t est le terme maximal de la classe e .*
- *Soit m tel que si $x \prec m$, alors $ext_E(x) \supset e$, m est un terme minimal de la classe e .*

Les termes minimaux sont ceux trouvés classiquement par les méthodes top-down. La question de savoir si un terme *minimal* est préférable à un terme *maximal* est cependant une question qui peut être différée après l'exploration : il est toujours possible de se ramener à un équivalent minimal lorsqu'on parcourt une espace réduit, ce que fait ELENA.

Cependant, dans certaines conditions les méthodes bottom-up ne sont pas satisfaisantes, principalement lorsqu'il est difficile de se diriger au début de l'exploration car aucune hypothèse ne semble meilleure qu'une autre (phénomène de *plateau*). Il en résulte que les choix de généralisations peuvent être malencontreux[Für02]. De plus si l'espace des termes candidats est réduit, chaque terme peut-être atteint par plusieurs généralisations de sous-ensembles d'exemples, par exemple si la taille de \mathcal{L} est réduite mais que $|E|$ est grand, ce qui peut rendre inefficace (en ressources) la recherche.

Ceci nous a amenés à considérer le parcours top-down d'un espace réduit en tant qu'espace quotient : l'idée est alors d'utiliser un opérateur de spécialisation garantissant de passer de classe d'équivalence en classe d'équivalence. Nous avons proposé un tel parcours de l'espace quotient relativement à \equiv_E en modifiant l'algorithme d'apprentissage top-down CN[CB91]. CN est un algorithme d'apprentissage supervisé multi-classe qui construit une liste de décision, c'est à dire un ensemble ordonné de règles associées chacune à un label de classe parmi n . Deux implémentations de CN explorant l'une l'espace original, l'autre l'espace quotient sont comparées dans [BS94] et les gains obtenus en temps et en nombre de termes explorés dans l'espace quotient sont mis en évidence expérimentalement, en particulier sur des problèmes ayant une théorie du domaine (cachée). Le lien est fait dans ce travail avec la notion de correspondance de Galois entre treillis d'instance et treillis de termes proposée par J-G. Ganascia [Gan93].

Marc Champesme a exploré cette notion de réduction dans le cadre de la programmation logique inductive [CBS95] et des graphes conceptuels [Cha95]. Le passage à l'ordre 1 engendre plusieurs difficultés : suivant la relation d'ordre utilisée, l'espace d'hypothèse peut être infini et les termes maximaux de très grande taille (sous θ -subsumption), ou encore non uniques (sous OI -subsumption). Par exemple, lorsque la *lgg* n'est pas unique, \mathcal{L} n'est pas un treillis et il n'y a plus de notion de fermeture. Néanmoins, il reste possible de parcourir l'espace quotient, bien que

plus difficilement puisqu'il y a plusieurs chemins d'une classe à une autre et qu'il faut préserver la complétude de l'opérateur de spécialisation.

Nous avons ensuite introduit un opérateur de spécialisation dans l'espace des termes fermés relativement à E^+ , c'est à dire l'espace quotient relativement à \equiv_{E^+} parcouru par les approches bottom-up comme ELENA. Pour traiter les problèmes multi-classes ELENA, contrairement à CN, utilise le fait que la stratégie *separate and conquer* pour l'apprentissage d'une hypothèse $h_1 \text{ OU } \dots \text{ OU } \dots h_k$ peut être efficacement étendue aux problèmes à n -classes selon l'approche *one vs all* [RK04] en construisant séparément des hypothèses pour $E^+ = E^1 \text{ vs } E^- = E^2 \cup \dots \cup E^n$, puis pour $E^+ = E^2 \text{ vs } E^- = E^1 \cup E^3 \cup \dots \cup E_n$, jusqu'à construire une hypothèse pour $E^+ = E^n \text{ vs } E^- = E^1 \cup \dots \cup E^{n-1}$. Pour chaque terme de chaque hypothèse, le programme TABATA [BS98b], explore de manière bi-directionnelle, guidée par une *recherche taboue* [Glo90], l'espace quotient relativement à E^+ : à chaque pas d'exploration, le voisinage dans l'espace quotient du terme courant t est constitué de ses moindres généralisés de la forme $t \wedge_{\mathcal{L}} x$ ainsi que de ses moindres spécialisés, dans l'espace réduit, de la forme $\text{int} \circ \text{ext}_{E^+}(t \vee_{\mathcal{L}} a)$, où a est un élément $\vee_{\mathcal{L}}$ -irréductible de \mathcal{L} . La méthode tabou consiste à toujours aller vers le meilleur voisin, même si celui-ci est moins bon que le terme d'origine, mais après un pas dans le voisinage l'élément atteint est mis sur une file tabou de taille limitée, ce qui permet d'échapper à certains optima locaux. TABATA a de très bons résultats, y compris sur des problèmes difficiles pour les méthodes top-down, comme la fonction XOR (sous le nom *even-parity*). Une autre observation porte sur la méthode de décision. En effet en conservant après l'apprentissage d'un problème n -classes les hypothèses H_1, \dots, H_n sous leur forme de prototypes, et en utilisant, une méthode de classification des instances de type *plus proche prototype voisin*, on peut, sur certains problèmes obtenir de forts gains en justesse sur des données test. Cette technique de classification utilisée par S. Salzberg [Sal91] et par P. Domingos [Dom96], a l'avantage de remplacer au moment de la classification les conflits (l'instance satisfait plusieurs hypothèses H_i) par des silences (l'instance ne satisfait aucune hypothèse H_i), ce qui permet d'utiliser la distance aux prototypes comme critère robuste de classification. Sur le problème waveform testé (3 classes), la solution H'_1, \dots, H'_3 , équivalente sur E à H_1, \dots, H_3 , mais dont les termes sont minimaux, obtient des justesses comparables à CN ou CART, et est dominée (de ≈ 0.11) par la solution de termes maximaux utilisant la notion de distance précitée.

Ces approches ont des liens avec les travaux en apprentissage supervisé utilisant les treillis de Galois (ou treillis de concepts) menés à la même époque, sous la direction de J. Sallantin à Montpellier, par M. Liquière et E. Mephu-Nguifo [LM90, NN97] et proposant des systèmes d'apprentissage par règles construisant la partie supérieure du treillis de Galois. Notre approche se différencie par l'introduction d'opérateurs de raffinement permettant de passer de fermés en fermés pendant une exploration classique de l'espace de termes, par opposition à l'exploration guidée par la construction partielle du treillis de Galois. Des travaux récents utilisent les treillis de Galois en apprentissage supervisé, comme l'élaboration d'un algorithme exact de construction d'arbre de décision [NF07], la représentation simultanée d'un ensemble d'arbres de décision [BBOV09], la constitution de l'architecture d'un réseau multicouche à partir des noeuds du treillis [TNT07] ou encore pour définir des ensembles de termes pertinents pour une tâche de discrimination [GKL08].

2.3 Fermés fréquents en fouille de données

Il est à remarquer que les énoncés fermés ont été récemment très étudiés en *Fouille de Données* : il y a maintenant de nombreux algorithmes qui recherchent les motifs fermés fréquents (par exemple LCM [UAUA04]), construisant ou non le treillis associé. Ces motifs fermés fréquents, sont beaucoup moins nombreux et représentent la même information du point de vue extensionnel que les motifs fréquents. D'un point de vue formel, le problème de la recherche des itemsets fermés se ramène à la recherche de cliques maximales bipartites dans un graphe, et donc de cliques maximales [ZO98, LLSW05]. D'un point de vue théorique et applicatif, la recherche de motifs fréquents, et fermés fréquents, s'étend actuellement à des formes de motifs diverses, séquences, graphes, arbres [PYW⁺08][TRS04] en raison de la prévalence actuelle de données semi-structurées. Remarquons toutefois que la formulation "closed pattern" se réfère en fouille de données, même dans le cas où le moindre généralisé n'est pas unique, à tout terme maximal dans sa classe d'équivalence extensionnelle [TRS04]. Enfin, pour conclure, des travaux récents

Chapitre 3

Abstractions Extensionnelles et Intensionnelles Ordonnées

Nous nous intéressons ici à la relation intension/extension avec comme préoccupation majeure de diminuer la taille de l'espace des couples intension/extension observés, tout en préservant l'ordre sur cet espace. Nous définissons cette réduction comme une opération d'*abstraction*, en ce sens que la finalité est de pouvoir induire ou raisonner en simplifiant, en fonction de certains *a priori*, les représentations. La proposition ci-dessous sera confrontée au paragraphe 3.1.6 aux modèles d'abstraction rencontrés en Apprentissage Artificiel.

3.1 Abstractions extensionnelles et intensionnelles

Je présente ici la notion d'abstraction d'une manière différente de celle proposée, d'une manière indirecte et sans utiliser ce mot, dans les travaux commentés aux paragraphes suivants [VS05a]. Dans ces derniers travaux, nous partons d'une notion de projection, également utilisée en Analyse de Concepts Formels, dans le cas intensionnel, par S. Kuznetsov [GK01]. Il s'agit, d'un point de vue général, d'une fonction d'un treillis vers une partie de celui-ci préservant l'ordre partiel, et formant elle-même un treillis. Une projection se définit comme une *fermeture*, à ceci près que l'élément projeté est cette fois plus petit que son antécédent (on parle aussi d'opérateur *d'ouverture*). En utilisant une projection sur un des treillis d'une correspondance de Galois, on change les notions d'intension ou d'extension, et on diminue le nombre de classes d'équivalences correspondantes. On aboutit ainsi à des représentations plus *grossières*. La conséquence, recherchée, est que le treillis de Galois obtenu, matérialisant la relation entre intension et extension, soit lui même le résultat de la projection du treillis de Galois original. Avant de décrire les propriétés des treillis obtenus, nous allons définir une notion d'abstraction, et montrer que les treillis *abstraits* obtenus sont les mêmes que les treillis obtenus par projection. L'intérêt de cette nouvelle formulation est qu'elle est constructive : ainsi définir une partie d'un langage qui soit le résultat d'une projection n'est pas immédiat, même si le vérifier est simple, alors qu'à partir de n'importe quel sous-ensemble du langage on induit une abstraction de celui-ci.

3.1.1 Projection d'un treillis et abstraction correspondante

Nous rappelons ci-dessous la notion générale de projection :

Définition 6 (Projection) *p est une projection d'un ensemble ordonné (M, \leq) ssi pour tout couple (x, y) d'éléments de M , on a :*

$x \geq p(x)$ (minimalité)

Si $x \leq y$ alors $p(x) \leq p(y)$ (monotonie)

$p(x) = p(p(x))$ (idempotence)

On remarque que \perp_P appartient nécessairement à $p(P)$ car pour tout élément x de P on a $p(x) \leq x$ et donc en particulier $p(\perp_P) \leq \perp_P$ et donc $p(\perp_P) = \perp_P$.

Nous définirons une *abstraction* de P comme suit :

Définition 7 (Abstraction d'un treillis) *Une abstraction d'un treillis P est un sous-ensemble de P , fermé pour \vee_P et contenant \perp_P .*

Par la suite nous considérons que $\bigvee_P \emptyset = \perp_P$ et donc que \perp_P fait partie de la fermeture par \vee_P d'une partie de P .

Une projection a aussi la propriété suivante[PRSV02] :

Proposition 8 (Treillis projeté) *Soit P un treillis avec les opérateurs \vee_P et \wedge_P et p une projection définie sur P , alors $p(P)$ est aussi un treillis avec les opérateurs \wedge et \vee définis comme suit :*

$$p1 \wedge p2 = p(p1 \wedge_P p2)$$

$$p1 \vee p2 = p1 \vee_P p2$$

L'opérateur \vee étant \vee_P , le projeté $p(P)$ d'un treillis P est donc une abstraction de P . Mais l'inverse est aussi vrai :

Proposition 9 (Projection associée à une partie C de P) *Tout sous-ensemble C d'un treillis P est associé à la projection p définie par $p(y) = \bigvee_{c \in C, c \leq y} c$. $p(P)$ est alors le treillis de la fermeture par \vee_P de C .*

Preuve Nous allons montrer que p est une projection de P .

1. $p(y) \leq y$? $p(y)$ est le plus petit majorant de l'ensemble d'éléments $c \leq y$ donc $p(y) \leq y$. Si $y = \perp_P$ alors l'ensemble des c tels que $c \leq y$ est vide et $p(y) = \perp_P$.
2. Soit $x \leq y$, $p(x) \leq p(y)$? $x \leq y$ implique que si $c \leq x$ alors $c \leq y$ donc $\bigvee_{c \in C, c \leq x} c \leq \bigvee_{c \in C, c \leq y} c$.
3. $pp(y) = p(y)$? Soit un élément $c \in C$ tel que $c \leq y$. On a alors $c \in \{q \in C \mid q \leq y\}$, donc $c \leq \bigvee_{q \in C, q \leq y} q = p(y)$. En conséquence $\bigvee_{c \in C, c \leq y} c \leq \bigvee_{c \in C, c \leq p(y)} c$ et donc $p(y) \leq p(p(y))$, et d'après 1 $p(p(y)) \leq p(y)$ et donc $pp(y) = p(y)$.

Par ailleurs pour tout c de C on a $c = p(c)$ car $c \in \{c_i \in C \mid c_i \leq c\}$. p étant une projection on a pour toute paire (c_1, c_2) d'éléments de C , $p(c_1 \vee c_2) = p(c_1) \vee p(c_2)$ et donc $= c_1 \vee c_2$. Toute disjonction d'éléments de C est donc dans $p(P)$ et tout élément de $p(P)$ étant par définition une disjonction d'éléments de C , la fermeture par \vee d'éléments de C est donc $p(P)$.

□

D'autre part pour obtenir une abstraction d'un treillis P il suffit de prendre un sous-ensemble C de P et de le fermer par \vee_P . Un corollaire évident est que nous avons mis en bijection les projections de P et les abstractions de P .

L'ensemble des résultats obtenus avec les projections concernant la relation entre intension et extension, que nous allons résumer au paragraphe suivant, s'interprètent donc en termes d'abstractions.

D'un point de vue pratique il nous sera utile de considérer l'ensemble des \vee -irréductibles d'une abstraction :

Proposition 10 *Soit O_A l'ensemble des éléments \vee -irréductibles d'une abstraction A , la projection p associée à O_A , $p(y) = \bigvee_{o \in O_A, o \leq y} o$ est la projection associée à A .*

Preuve Comme la fermeture par \vee_P de O_A est A , d'après la proposition 9, p est associée à A . □

Un cas particulier intéressant est le cas où $O_A \subseteq O$: pour obtenir A , on prend une partie des éléments \vee -irréductibles et on les ferme par \vee_P . Nous dirons alors que A est une abstraction *directe* de P .

Nous considérons par la suite deux types de projections : une projection *intensionnelle* est une projection sur \mathcal{L} , et une projection *extensionnelle* est une projection sur $\mathcal{P}(I)$.

Abstractions et projections intensionnelles Nous prenons ci-dessous quelques exemples de projection intensionnelles directes intuitives :

1. L'attribut hiérarchique a donné en 2.2.1 et représenté Figure 2.2 est une forme et on considère le treillis de valeurs $T_a = \{\text{rectangle}, \text{losange}, \text{triangle}, \text{quadrilatère}, \top_a = \text{forme impossible}, \perp_a = \text{forme}\}$ où *quadrilatère* représente *rectangle* $\wedge_{\mathcal{L}_a}$ *losange* et *forme* représente *rectangle* $\wedge_{\mathcal{L}_a}$ *losange* $\wedge_{\mathcal{L}_a}$ *triangle*. Tous les éléments de \mathcal{L}_a sont \vee -irréductibles sauf \perp_a et \top_a . Pour une hiérarchie toute abstraction est directe.
 - Une manière intuitive d'abstraire une hiérarchie est de ne considérer que les niveaux les plus généraux : on transforme des noeuds intérieurs en feuilles (leurs successeurs sont éliminés). Ici la racine est \perp_b et nous appellerons *feuille* un élément voisin de \top_b : en transformant *quadrilatère* en feuille on obtient $O'_a = \{\text{triangle}, \text{quadrilatère}\}$.
 - En considérant la partie $C = \{\text{rectangle}, \text{losange}, \text{triangle}\}$ on obtient l'abstraction $A = \{\top_a, \text{triangle}, \text{rectangle}, \text{losange}, \perp_a\}$. Dans la nouvelle hiérarchie le niveau *quadrilatère* a disparu. La représentation est simplifiée mais cela peut sembler contre-intuitif de parler d'abstraction dans la mesure où ce que l'on abandonne en faisant une abstraction est souvent plus spécifique que ce que l'on garde. Cependant l'idée ici est, comme nous allons le voir aux paragraphes suivants, que l'ensemble des couples intension/extension obtenus avec une abstraction est un sous-ensemble de ceux obtenus sans abstraction.
2. Un treillis fini d'intervalles $\mathcal{L}_b = \{[x_i, x_j] \mid x_i, x_j \in \{x_0, \dots, x_n\}, x_i < x_j\}$ est représenté par l'ensemble de ses bornes et ses éléments \vee -irréductibles sont de la forme $[x_i, x_n]$ ou $[x_0, x_j]$. En apprentissage supervisé on considérera comme bornes les *valeurs-frontières* de sorte que les exemples soient de même label dans tout intervalle. De ce fait toute l'information, en ce

qui concerne la séparation des exemples en labels, est préservée. Par la suite une méthode de discrétisation sélectionnera un sous-ensemble de ces bornes de sorte à satisfaire un certain critère [FI93]. La sélection de bornes correspond à une sélection d'éléments \vee -irréductibles, et conduit à une abstraction directe B . Sélectionner des bornes ramène au cas d'un treillis d'intervalles entiers : chaque entier i est associé à un intervalle $]x_{i-1}, x_i]$, et ainsi $]x_i, x_j]$ est associé à l'intervalle entier $[i+1 \dots j]$. Si nous reprenons par exemple le treillis des intervalles inclus dans $[1..4]$ de la figure 2.2, les éléments \vee -irréductibles sont $[1..4]$, $[1..3]$, $[1..2]$, $[1]$, $[2..4]$, $[3..4]$, $[4]$.

3. De nombreux algorithmes d'apprentissage à base de règles utilisent à chaque recherche d'un terme de la stratégie de couverture un exemple positif particulier g , nommé *graine*, et restreignent \mathcal{L} au langage \mathcal{L}_g des termes t tels que $g \in \text{ext}(t)$, c-à-d $\dot{g} \succeq t$. \mathcal{L}_g est une abstraction directe obtenue en sélectionnant les éléments \vee -irréductibles plus généraux que \dot{g} . Dans l'exemple précédent, si $\dot{g} = [3]$, on obtient le treillis engendré par les éléments \vee -irréductibles $[1..3]$, $[2..4]$, $[3..4]$. Une abstraction intensionnelle peut donc faire intervenir l'extension dans sa définition.
4. Le langage $\mathcal{L} = \mathcal{L}_a \times \mathcal{L}_b$ est lui-même abstrait en $A \times B$. De plus, \mathcal{L} a comme éléments \vee -irréductibles les couples d'éléments \vee -irréductibles de \mathcal{L}_a et \mathcal{L}_b . On en déduit que les abstractions directes de \mathcal{L} sont les produits d'abstractions directes de \mathcal{L}_a et \mathcal{L}_b .

La figure 3.1 présente les abstractions de hiérarchie et d'arbres mentionnées ci-dessus. On re-

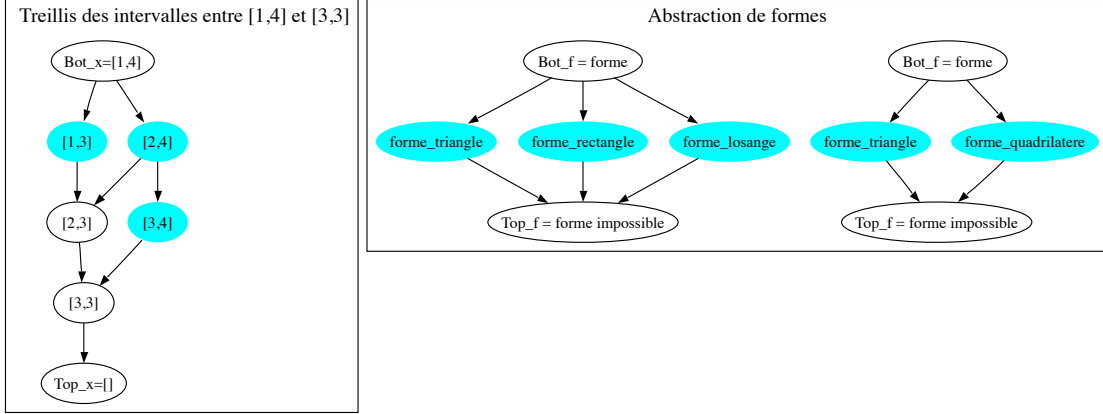


FIGURE 3.1 – À gauche l'abstraction du treillis d'intervalles de la figure 2.2 obtenue en se restreignant aux intervalles contenant la valeur de la graine $g = [3, 3]$. À droite les deux abstractions de la hiérarchie de la figure 2.2 obtenues en éliminant le niveau de représentation intermédiaire *quadrilatère* (à gauche) et en transformant le noeuds intérieur *quadrilatère* en feuille (à droite). Dans le premier cas, la dénomination "abstraction" peut être contre-intuitive.

marquera que certains sous-langages sont inaccessibles par abstraction intensionnelle alors même qu'ils correspondent intuitivement à une abstraction. Dans l'exemple du treillis d'intervalles précédent, un treillis intuitivement plus abstrait serait obtenu en éliminant les termes les plus spécifiques $[1]$, $[2]$, $[3]$, $[4]$, $[5]$. Le treillis engendré par \vee -fermeture de $\mathcal{L} - \{[1], [2], [3], [4], [5]\}$ contient ces éléments. Il ne s'agit donc pas d'une abstraction intensionnelle telle qu'elle est définie ici. Intuitivement

l'obstacle ici est que l'on a supposé que pour toute instance x , son abstraction \hat{x} était unique, ce qui est une condition permettant de garantir l'unicité également de toutes les généralisations maximales spécifiques $\text{int}(e)$ pour tout sous-ensemble d'instance e . Lorsqu'on élimine ici $\{[1], [2], [3], [4], [5]\}$, toute instance x ayant une valeur $b = i$ aura deux intervalles minimaux $[i - 1, i]$ et $[i, i + i]$ le contenant et donc \hat{x} n'est plus unique.

Abstractions et projections extensionnelles En ce qui concerne les projections extensionnelles, et les abstractions extensionnelles correspondantes, elles s'interprètent en considérant que l'ensemble des parties $\mathcal{P}(I)$ d'un ensemble d'instances $I = \{i_1, i_2, \dots, i_n\}$ est la fermeture par réunion des singletons $\{\{i_1\}, \{i_2\}, \dots, \{i_n\}\}$. On remarque que ces singletons sont aussi les éléments \cup -irréductibles du treillis booléen $\mathcal{P}(I)$. Une abstraction extensionnelle, en ne considérant qu'une partie $E \subseteq \mathcal{P}(I)$ fermée par réunion, limite les extensions aux éléments de E . Si on considère les éléments \cup -irréductibles de E , comme le suggère la proposition 10, on obtient un *recouvrement partiel* de I qui est caractéristique de E : un ensemble de parties de I qui permettent d'engendrer par réunion E . Ces éléments sont caractéristiques du changement de granularité sur l'ensemble des instances : les instances sont abstraites en *groupes d'instances*. Nous donnons ci-dessous quelques exemples d'abstractions extensionnelles sur $I = \{1, 2, 3\}$ et les projections associées :

- Enlever un élément de I :

$$\begin{aligned} p_1(\{1, 2, 3\}) &= p_1(\{1, 2\}) = \{1, 2\} \\ p_1(\{1, 3\}) &= p_1(\{1\}) = \{1\} \\ p_1(\{2, 3\}) &= p_1(\{2\}) = \{2\} \\ p_1(\{3\}) &= p_1(\emptyset) = \emptyset \end{aligned}$$
- Ne retenir que les parties d'au moins 2 éléments de $\mathcal{P}(I)$:

$$p_2(e) = e$$
- Sauf :

$$p_2(\{1\}) = p_2(\{2\}) = p_2(\{3\}) = \emptyset$$
- Engendrer les unions des *groupes* $\{1, 2\}$ et $\{2, 3\}$:

$$p_3(e) = e$$
- Sauf :

$$\begin{aligned} p_3(\{1\}) &= p_3(\{2\}) = p_3(\{3\}) = \\ p_3(\{1, 3\}) &= \emptyset \end{aligned}$$
- Engendrer un emboîtement de *groupes d'instances* :
 - La fermeture par réunion de $P = \{\{1\}, \{1, 2\}, \{1, 2, 3\}\}$ est $E = \{\emptyset, \{1\}, \{1, 2\}, \{1, 2, 3\}\}$, dont les éléments \cup -irréductibles sont précisément $O_E = P$.
 - On obtient alors la projection $p_4 : p_4(e) = \bigcup_{o \in P, o \subseteq e} o$. On a ainsi :

$$\begin{aligned} p_4(e) &= e \text{ pour } e \in E \\ p_4(\{2, 3\}) &= p_4(\{3\}) = p_4(\{2\}) = \emptyset \end{aligned}$$

Les abstractions correspondantes sont présentées Figure 3.2.

Dans la suite, il sera question des treillis Alpha. Ceux-ci sont obtenus en généralisant l'abstraction correspondant à la sélection des sous-ensembles *fréquents* de I^1 . L'idée est de partir d'un ensemble de *classes* \mathcal{C} , puis de considérer les éléments de $\mathcal{P}(I)$ constitués de parties fréquentes relativement à chaque classe. Le changement de granularité est alors, d'une part, dépendant d'un

1. On remarquera que la réunion de deux ensembles fréquents (c'est-à-dire dont le cardinal est supérieur à un seuil fixé) est un ensemble fréquent.

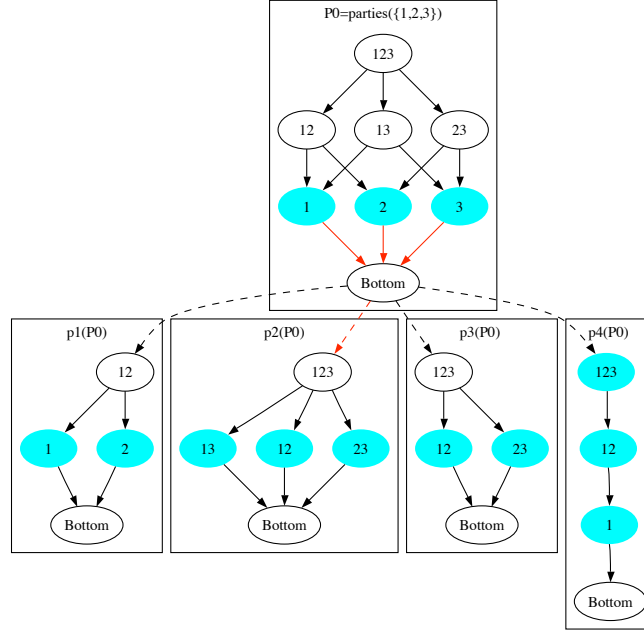


FIGURE 3.2 – Le treillis des parties de $\{1, 2, 3\}$ et quelques abstractions. Dans chaque cas les éléments \cup -irréductibles (en bleu ou gris) suffisent, lorsqu'on les ferme par réunion, à engendrer l'abstraction.

point de vue sur les instances : leur regroupement en classes éventuellement recouvrantes, et d'autre part, d'une notion de fréquence locale. Plus précisément :

Définition 8 (Abstraction Alpha) Soit \mathcal{C} un recouvrement partiel en classes de I . On lui associe, pour $\alpha \in [0, 1]$ donné, le treillis $E_\alpha \subseteq \mathcal{P}(I)$ défini ci-dessous :

On considère les plus petits groupes d'instances fréquents dans chaque classe :

$$\mathcal{C}_\alpha = \{u \mid \exists u_j \in \mathcal{C} \text{ tel que } u \subseteq u_j \text{ et } |u| \geq \alpha \cdot |u_j| \text{ et } u' \subset u \Rightarrow |u'| < \alpha \cdot |u_j|\}$$

L'abstraction Alpha E_α est obtenue en fermant par réunion \mathcal{C}_α .

La projection correspondante

$$p_{\mathcal{C}_\alpha}(e) = \bigcup_{u \in \mathcal{C}_\alpha, u \subseteq e} u$$

s'écrit aussi :

$$p_{\mathcal{C}_\alpha}(e) = \{i \in e \mid \exists u_j \in \mathcal{C} \text{ tel que } i \in u_j \text{ et } |e \cap u_j| \geq \alpha \cdot |u_j|\}$$

Cette dernière écriture permettra un calcul beaucoup plus efficace des éléments de E_α .

Les cas particuliers intéressants sont $E_0 = \mathcal{P}(I)$ (pas d'abstraction), E_1 dont tout élément est une réunion de classes de \mathcal{C} , et enfin le cas $\mathcal{C} = \{I\}$ qui correspond aux parties fréquentes (*i.e.* de cardinal supérieur à $\alpha \cdot |I|$) de I .

Dans l'exemple de la Figure 3.3 sont représentés d'abord le recouvrement $\mathcal{C} = \{\{1, 2, 3\}, \{4, 5, 6\}\}$ et ses deux abstractions E_1 et $E_{2/3}$, puis l'abstraction fréquente ($\alpha = 1/2$) pour $\mathcal{C} = \{\{1, 2, 3, 4\}\}$.

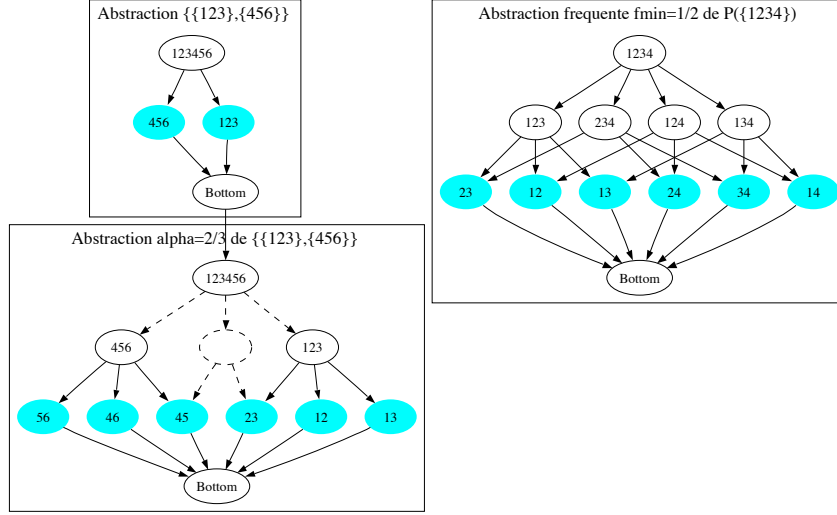


FIGURE 3.3 – A droite l’abstraction fréquente ($\alpha = 1/2$) pour $\mathcal{C} = \{\{1, 2, 3, 4\}\}$: l’ensemble des parties de 2 éléments de $\{1, 2, 3, 4\}$. A gauche deux abstractions E_1 (en haut) et $E_{2/3}$ (en dessous) pour le recouvrement $\mathcal{C} = \{\{1, 2, 3\}, \{4, 5, 6\}\}$. Dans $E_{2/3}$ certains éléments obtenus par réunion des éléments de $\mathcal{C}_{2/3}$ ne sont pas représentés (la partie absente est suggérée par des pointillés) comme par exemple $\{2, 3, 4, 5\} = \{2, 3\} \cup \{4, 5\}$.

De manière plus générale on peut définir une projection extensionnelle relativement à un treillis extensionnel E , où il est entendu que E est lui-même le résultat d’une projection de $P(I)$, ce qui permet de composer les projections extensionnelles et qui sera utile par la suite pour ordonner les treillis.

Enfin, un cas particulier important est celui où $A = p(E)$ est fermé par \wedge_E , c’est-à-dire où $p(E)$ est un sous-treillis de E . Dans ce cas on a en effet $\wedge_A = \wedge_E$ et donc $p(a \wedge_E b) = p(a) \wedge_E p(b)$, et si $a, b \in A$, alors $p(a \wedge_E b) = a \wedge_E b$.

Proposition 11 *A est une abstraction de E fermée par \wedge_E , ssi la projection associée p_A est telle que*

$$\forall e, e' \in A, p_A(e \wedge'_E e') = e \wedge_E e'$$

Exemple Considérons la hiérarchie \mathcal{L} de la Figure 2.2. Considérons la partie $A_1 = \mathcal{L} - \{\text{quadrilatere}\}$. A_1 est une abstraction (au centre Figure 3.1), cependant A_1 n’est pas fermée par $\wedge_{\mathcal{L}}$ ($\text{losange} \wedge_{\mathcal{L}} \text{rectangle} = \text{quadrilatere}$ n’est pas dans A_1). En revanche $A_2 = \mathcal{L} - \{\text{losange}, \text{rectangle}\}$ (à droite Figure 3.1) est fermée par $\wedge_{\mathcal{L}}$ et est une abstraction plus intuitive. \square

En particulier pour les abstractions extensionnelles, en prenant $E = \mathcal{P}(I)$ cela signifie que $\text{ext}_A(a \vee_{\mathcal{L}} b) = \text{ext}_A(a) \cap \text{ext}_A(b)$ d’après la propriété 6 de la proposition 4 ce qui évite, pendant une exploration de \mathcal{L} du plus général au plus spécifique, de calculer les projections, sauf pour les éléments $\vee_{\mathcal{L}}$ -irréductibles.

En ce qui concerne les abstractions extensionnelles, un sous-ensemble T de $\mathcal{P}(I)$ fermé par intersection et réunion est une *topologie*, (I, T) est un *espace topologique* et les éléments de T sont appelés des *ouverts*. On peut engendrer une topologie T en considérant une *base d'ouverts* appelés aussi *granules ouverts*, c'est-à-dire un sous-ensemble B de $\mathcal{P}(I)$, recouvrant I^2 , et tel que toute intersection de 2 éléments B_i et B_j de B s'écrit comme une réunion d'éléments de B . L'ensemble T est alors obtenu par fermeture par réunion de B et est également fermé par intersection [CC08]. La technique peut-être étendue à tout treillis fini E distributif, nous parlerons alors d'*abstraction topologique*.

Proposition 12 *Soient un treillis distributif fini E et un sous-ensemble \mathcal{B} tel que $\forall B_i, B_j \in \mathcal{B}, B_i \wedge_E B_j$ s'écrit comme la disjonction \vee_E d'éléments de B . Alors ;*

La fermeture A par \vee_E de B est aussi fermée par \wedge_E et A est distributif.

Preuve Soit X et Y deux éléments de l'abstraction A associée à \mathcal{C} . On a donc :

$X \wedge_E Y = (B_{i_1} \vee_E \cdots \vee_E B_{i_k}) \wedge_E (B_{j_1} \vee_E \cdots \vee_E B_{j_m})$. Par distributivité on obtient :

$X \wedge_E Y = (B_{i_1} \wedge_E B_{j_1}) \vee_E \cdots \vee_E (B_{i_k} \wedge_E B_{j_m})$. Comme chaque paire $(B_x \wedge_E B_y)$ s'écrit $B_{l_1} \vee_E \cdots \vee_E B_{l_r}$, on en déduit que $X \wedge_E Y$ s'écrit comme une disjonction par \vee_E d'éléments de \mathcal{B} , et donc A est fermée par \wedge_E . \square

En pratique dans le cas extensionnel on part d'un recouvrement partiel \mathcal{C} de $\mathcal{P}(I)$, on le ferme par intersection en cherchant un point fixe aux ensembles des intersections d'éléments de \mathcal{C} , puis on le ferme par réunion.

Exemple Soient $I = \{1, 2, 3, 4, 5, 6, 7\}$ et $\mathcal{C} = \{C_1 = \{1, 4, 5, 7\}, C_2 = \{4, 2, 7, 6\}, C_3 = \{5, 7, 6, 3\}\}$. On construit d'abord $C_1 \cap C_2 = \{4, 7\}$, $C_2 \cap C_3 = \{7, 6\}$, $C_1 \cap C_3 = \{5, 7\}$, puis $C_1 \cap C_2 \cap C_3 = \{7\}$. Ce nouveau recouvrement engendre une abstraction fermée par intersection qui n'est pas une partition. Remarquons que si \mathcal{C} est une partition, alors l'abstraction associée est fermée par intersection. \square

3.1.2 Ordre sur les abstractions

On définit un ordre partiel sur les projections, et donc de manière équivalente sur les abstractions, de la manière suivante :

Définition 9 (Ordre sur les projections)

Soit un treillis E, \vee_E, \wedge_E, p_1 et p_2 deux projections définies sur E , nous dirons que $p_2 \leq p_1$ (p_2 est plus grossière (moins fine) que p_1) ssi

- *Il existe une projection p définie sur $p_1(E)$ telle que pour tout e de E , $p_2(e) = p \circ p_1(e)$ ou de manière équivalente :*
- *Tout élément \vee -irréductible de $p_2(E)$ est la disjonction d'éléments \vee -irréductibles de $p_1(E)$.*

On pourrait ci-dessus remplacer \vee -irréductible de $p(E)$ par *générateur* de $p(E)$: un sous-ensemble de $p(E)$ dont la fermeture par \vee est égale à $p(E)$ ³. En pratique cela signifie que si

2. Dans notre notion d'abstraction le recouvrement peut n'être que partiel, mais il suffit alors de considérer $p(I)$ comme ensemble de départ.

3. Un générateur contient donc tous les éléments \vee -irréductibles.

l'on prend deux sous-ensembles G_1 et G_2 de E , ils sont alors générateurs d'abstractions $p_1(E)$ et $p_2(E)$ telles que $p_2(E)$ est plus grossière que $p_1(E)$ ssi tout élément de G_2 peut s'écrire comme disjonction d'éléments de G_1 .

Exemple Soit $E = \mathcal{P}(\{1, 2, 3, 4\})$, $G_1 = \{\{1, 2\}, \{1, 3\}\}$ et $G_2 = \{\{1, 2\}, \{1, 2, 3\}\}$, l'abstraction engendrée par G_2 est plus grossière que l'abstraction engendrée par G_1 , puisque $\{1, 2\} \in G_1$ et $\{1, 2, 3\} = \{1, 2\} \cup \{1, 3\}$. \square

Cet ordre partiel défini lui-même un treillis. On trouvera Figure 3.4 les bornes supérieures et inférieures de deux projections du treillis $E = \mathcal{P}(\{1, 2, 3, 4\})$.

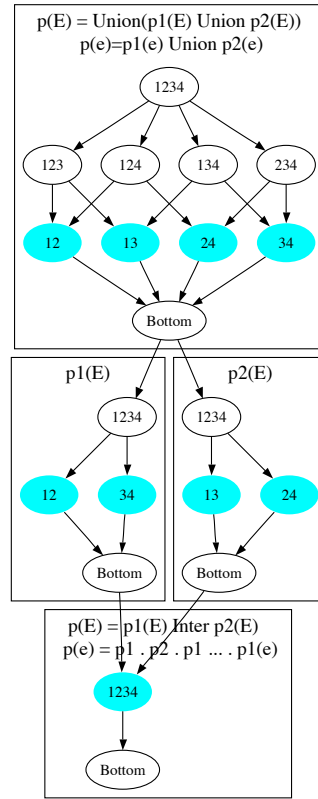


FIGURE 3.4 – Extrait du treillis des abstractions de $E = \mathcal{P}(\{1, 2, 3, 4\})$. La borne supérieure de deux abstractions $p_1(E)$ et $p_2(E)$ est la fermeture par réunion de $p_1(E) \cup p_2(E)$. La borne inférieure est simplement $p_1(E) \cap p_2(E)$. On remarque que la projection associée à $p_1(E) \cup p_2(E)$ est $p(e) = p_1(E) \cup p_2(e)$ mais que celle associée à $p_1(E) \cap p_2(E)$ est définie par la composition d'une suite alternée finie $p_1 \circ p_2 \circ \dots \circ p_1(e)$.

Proposition 13 (Treillis des abstractions) Soient un treillis E, \vee_E, \wedge_e et deux abstractions $p_1(E)$ et $p_2(E)$,

- Le plus petit majorant $p_1(E) \vee p_2(E)$ de $p_1(E)$ et de $p_2(E)$ est la fermeture par \vee_E de $p_1(E) \cup p_2(E)$. La projection correspondante est $p(e) = p_1(e) \vee_E p_2(e)$.
- Le plus grand minorant $p_1(E) \wedge p_2(E)$ de $p_1(E)$ et de $p_2(E)$ est simplement $p_1(E) \cap p_2(E)$. La projection correspondante est $p(e) = p_1 \circ p_2 \circ \dots \circ p_1(e)$.

Preuve Ébauche de preuve :

- Le plus grand minorant $p_1(E) \wedge p_2(E)$ de $p_1(E)$ et de $p_2(E)$ est simplement $p_1(E) \cap p_2(E)$. En effet si e et e' sont des éléments de $p_1(E) \cap p_2(E)$ alors ils sont éléments de $p_1(E)$ et donc $e \vee_E e'$ est élément de $p_1(E)$. De même pour $p_2(E)$. On en conclut que e et e' sont éléments de $p_1(E) \cap p_2(E)$. En revanche, soit e'' un élément de $p_1(E) - p_2(E)$, alors il n'appartient pas à une abstraction moins fine que $p_1(E)$ et moins fine que $p_2(E)$.
- $p_1(E) \vee p_2(E)$ doit contenir les éléments de $p_1(E)$ et ceux de $p_2(E)$, mais il doit par définition contenir aussi la disjonction de toute paire d'éléments de $p_1(E)$ et de $p_2(E)$.
- La projection correspondante à $p_1(E) \cap p_2(E)$, s'écrit $p(e) = p_1 \circ p_2 \circ \dots \circ p_1$ car pour être dans l'intersection, un élément doit se projeter sur lui-même selon p_1 et selon p_2 , toutes deux idempotentes. Il faut donc projeter successivement un élément selon p_1 et p_2 jusqu'à ce qu'une des projections soit égale à son argument, l'idempotence des projections assure alors qu'on trouve un point fixe. D'autre part les projections sont strictement décroissantes sauf si la projection est égale à son argument, auquel cas on est arrivé au point fixe, et donc celui-ci, au pire \perp_E , est atteint, E étant fini.

□

3.1.3 Treillis de Galois Abstraits et ordre sur les treillis de Galois abstraits

Bien qu'on puisse définir un treillis de Galois projeté directement relativement à deux projections extensionnelle p_e et intensionnelle p_i , nous considérons les deux cas séparément, de manière à garder une interprétation aussi intuitive que possible de ces changements de granularité. En réalité p_e et p_i se composent et commutent, il est donc équivalent de projeter successivement selon les deux ordres. Dans un contexte différent, celui de fonctions de cohérences, cela a été indépendamment remarqué par Edwin Diday[Did01]. Par ailleurs la notion de correspondance de Galois étant symétrique relativement aux deux ensembles, il ne s'agit ci-dessous que d'un seul résultat, décliné en deux théorèmes[PRSV02] :

Théorème 1 (projection intensionnelle d'un Treillis de Galois) Soient $(\mathcal{L}, ext, E, int)$ une correspondance de Galois, G le treillis de Galois associé, p_i une projection de \mathcal{L} ,

et notons $\mathcal{L}' = p_i(\mathcal{L})$, le langage projeté et $int' = p_i \circ int$ la fonction d'intension correspondante, on a alors :

1. $(\mathcal{L}', ext, E, int')$ est une correspondance de Galois, et on note G' le treillis de Galois associé.
2. Soit $(c, e = ext(c))$ un élément de G alors $(p_i(c), ext(p_i(c)))$ appartient à G' .

Nous dirons que $G' = p_i(G)$ est emboîté intensionnellement dans G par p_i .

Théorème 2 (projection extensionnelle d'un Treillis de Galois) Soient $(\mathcal{L}, ext, E, int)$ une correspondance de Galois, G le treillis de Galois associé, p_e une projection de E , et notons $E' = p_e(E)$, l'ensemble extensionnel projeté et $ext' = p_e \circ ext$ la fonction d'extension correspondante, on a alors :

1. $(\mathcal{L}, ext', E', int)$ est une correspondance de Galois, et on note G' le treillis de Galois associé.

2. Soit $(c = \text{int}(e), e)$ un élément de G alors $\text{int}(p_e(e), p_e(e))$ appartient à G' .
Nous dirons que $G' = p_e(G)$ est emboîté extensionnellement dans G par p_e .
3. La relation d'équivalence sur les termes $\equiv_{\text{ext}'}$ est plus grossière que \equiv_{ext} .

L'emboîtement est donc un ordre partiel sur les treillis de Galois, suivant l'ordre des projections (et donc abstractions) intensionnelles et extensionnelles correspondantes. Le nombre de noeuds est décroissant selon cet ordre. Remarquons que dans le cas des abstractions extensionnelles, on a pour tout terme t de \mathcal{L} , lorsque p_2 est plus grossière que p_1 , $\text{ext}_2(t) = p \circ \text{ext}_1(t)$ et donc $\equiv_{\mathcal{L}}^2$ plus grossière que $\equiv_{\mathcal{L}}^1$.

3.1.4 Extension et implication abstraites

Nous avons étudié [VS05a, VS05b], le treillis de Galois des abstractions Alpha définies au paragraphe 3.1.1 et en particulier les implications logiques et règles d'associations associées à l'inclusion des α -extensions. Les extensions Alpha sont un cas particulier des extensions abstraites définies ci-dessous :

Définition 10 Soit une abstraction extensionnelle A et sa projection associée p_A , Pour tout terme t de \mathcal{L} ,

$$\text{ext}_A(t) = p_A \circ \text{ext}(t)$$

est appelée l'extension abstraite de t .

Soit O_A l'ensemble des éléments \cup -irréductibles de A , on a aussi

$$\text{ext}_A(t) = \bigcup_{u \in O_A, u \subseteq \text{ext}(t)} u$$

On notera par la suite $O_A(i)$ l'ensemble des éléments de O_A auxquels appartient l'instance i . $O_A(i)$ forme l'ensemble des abstractions de l'instance i . Formellement $O_A(i) = \{u \in O_A \mid i \in u\}$.

L'inclusion d'extensions abstraites conduit à une notion d'implication abstraite :

Définition 11 (Implication abstraite) Soient deux éléments t_1 et t_2 de \mathcal{L} , on appellera implication abstraite, l'inclusion des extensions abstraites de t_1 et t_2 :

$$t_1 \rightarrow_A t_2 \text{ ssi } \text{ext}_A(t_1) \subseteq \text{ext}_A(t_2)$$

Du fait que cette notion d'implication découle de l'inclusion ensembliste elle est monotone, transitive et additive :

Proposition 14

Si $t_1 \rightarrow_A t_2$ et $t_2 \rightarrow_A t_3$, alors $t_1 \rightarrow_A t_3$

Si $t_1 \rightarrow_A t_2$, et $t_1 \leq_{\mathcal{L}} t$, alors $t \rightarrow_A t_2$

Si $t_1 \rightarrow_A t_2$ et $t_3 \rightarrow_A t_4$, alors $t_1 \vee_{\mathcal{L}} t_3 \rightarrow_A t_2 \vee_{\mathcal{L}} t_4$.

Une forme de modus ponens comme règle d'inférence est aussi valide :

Si $i \in \text{ext}_A(t_1)$ et $t_1 \rightarrow_A t_2$, alors $i \in \text{ext}_A(t_2)$

L'additivité au niveau des instances s'écrit :

Si $i \in \text{ext}_A(t_1)$ et $i \in \text{ext}_A(t_2)$ et $\exists u \in O_A(i)$, t.q. $u \subseteq \text{ext}(t_1 \vee_{\mathcal{L}} t_2)$, alors $i \in \text{ext}_A(t_1 \vee_{\mathcal{L}} t_2)$.

Les conditions sont donc réunies pour faire de la déduction monotone selon un certain degré de granularité extensionnelle.

Voici un exemple avec des α -extensions, en remarquant que si $\alpha_1 \leq \alpha_2$ pour le même recouvrement \mathcal{C} alors $p_{\mathcal{C},\alpha_1}$ est plus grossière que $p_{\mathcal{C},\alpha_2}$ (on écrit simplement $p_{\mathcal{C}}$ pour $p_{\mathcal{C},1}$) :

Exemple

$\mathcal{C} = \{\{\text{cheval, ecureuil-volant, platypus}\}, \{\text{canari, pie, poule}\}, \{\text{fourmi,abeille}\}\}.$

– $\text{ext}_{\mathcal{C}}(\text{ovipare}) = \{\text{ornithorynque, canari, pie, poule, fourmi,abeille}\}$

$\text{ext}_{\mathcal{C}}(\text{ovipare}) = \text{ext}_{\mathcal{C}_{2/3}}(\text{ovipare}) = \text{ext}_{\mathcal{C}_{1/3}}(\text{ovipare}) =$
 $\{\text{canari, pie, poule, fourmi,abeille}\}$

– $\text{ext}_{\mathcal{C}}(\text{vole}) = \{\text{écureuil-volant, canari, pie, abeille}\}$

$\text{ext}_{\mathcal{C}}(\text{vole}) = \emptyset$

$\text{ext}_{\mathcal{C}_{1/3}}(\text{vole}) = \{\text{canari, pie, abeille}\}$

$\text{ext}_{\mathcal{C}_{2/3}}(\text{vole}) = \{\text{canari, pie}\}$

– "Les animaux qui volent sont ovipares" pour $\mathcal{C}_{1/3}$ et $\mathcal{C}_{2/3}$

$\text{vole} \xrightarrow{\mathcal{C}_{1/3}} \text{ovipare est valide}$

$\text{vole} \xrightarrow{\mathcal{C}_{2/3}} \text{ovipare est valide}$

□

On remarque sur cet exemple que si on abstrait grossièrement, avec une catégorisation en mammifère, oiseau, insectes, on ne garde que les propriétés communes à tous les animaux d'une catégorie. En revanche l'utilisation d'un α assez grand permet de garder les propriétés qui sont communes à une proportion suffisante d'animaux de chaque classe, ce qui permet de faire apparaître l'implication "Les animaux qui volent sont ovipares" bien qu'ici l'écureuil volant (seul animal volant dans sa catégorie) contredirait l'implication classique, (correspondant à l'inclusion des extensions classiques). C'est donc bien une implication approximative, cependant transitive, et permettant donc le raisonnement déductif monotone qui est utilisée ici.

3.1.5 Règles d'associations abstraites

Les règles d'associations sont classiquement construites à partir d'un ensemble de termes fermés fréquents [PBTL99, Zak00]. Intuitivement un terme fermé est le représentant maximalement spécifique d'une classe d'équivalence de termes relative à l'extension, et a donc la même extension que les termes minimalement spécifiques de cette même classe, appelés *générateurs*. On en déduit pour chaque classe d'équivalence (associée à un noeud du treillis de Galois) un ensemble d'implications dont l'antécédent est un générateur, et le conséquent est le terme fermé. L'ensemble de ces règles forme la base *min-max* [PTB⁺05]. Cette base est étendue aux règles non exactes (de confiance différente de 1) liant les générateurs d'une classe aux fermés d'une autre. Les règles non exactes lient les générateurs d'une classe d'équivalence aux fermés d'une autre classe d'extension inférieure.

Nous adaptons ici la définition des règles d'association, définies sur l'extension classique, aux

extensions abstraites. Ici encore nous généralisons dans ce paragraphe, la notion de règles de α -association proposée et expérimentée en [VS05a, VS05b] et qui elle-même étend la notion classique de règle d'association. Chaque règle a un *support* (la fréquence observée sur les instances I de l'*antécédent* de la règle) et une *confiance* (celle-ci est inférieure à 1, valeur atteinte pour les implications valides sur I).

Définition 12 Une règle d'association relative à une abstraction A lie une paire de termes t_1 and t_2 . Le support et la confiance d'une règle d'association abstraite $r = t_1 \rightarrow_A t_2$ sont définies comme suit :

$$\begin{aligned} - A\text{-supp}(r) &= \frac{|ext_A(t_1 \vee_{\mathcal{L}} t_2)|}{|I|} \text{ and} \\ - A\text{-conf}(r) &= \frac{|ext_A(t_1 \vee_{\mathcal{L}} t_2)|}{|ext_A(t_1)|} \end{aligned}$$

La règle d'association abstraite $r = t_1 \rightarrow_A t_2$ est valide sur (I, A) lorsque $A\text{-supp}(r) \geq \text{minsupp}$ et $A\text{-conf}(r) \geq \text{minconf}$.

L'adaptation de la méthode de Pasquier et collaborateurs pour calculer les bases de règles min-max à partir des termes fermés et des termes générateurs est directe. Il s'agit essentiellement de remplacer le calcul des extensions par celui des A-extensions.

3.1.6 Abstraction ?

La notion d'abstraction a été souvent abordée en Intelligence Artificielle, dans différents contextes, et en particulier récemment, en Apprentissage. Afin de situer ce que j'ai appelé ici abstractions intensionnelles et extensionnelles, je me référerai d'abord ici à une présentation par J-D. Zucker des différents cadres définis pour aborder l'abstraction [Zuc03], ainsi qu'à la proposition par J-D. Zucker et L. Saitta du modèle KRA [SZ98][SZ07]. Dans ce modèle un cadre de représentation \mathcal{R} ⁴ d'un *monde perçu* est donné incluant plusieurs couches, de la perception dans une représentation \mathcal{P} , à un langage \mathcal{L} d'assertions sur le monde en question. Le monde perçu est éventuellement constitué de parties, par exemple une image observée décomposée en pixels, et le langage \mathcal{L} est d'ordre 1. Un opérateur d'abstraction, à un niveau donné, fait alors passer d'un langage à un langage *abstrait* différent dans le même formalisme. L'abstraction est fondée sur le niveau le plus bas, celui de la perception, et la propagation de contraintes de compatibilité vers les niveaux supérieurs. Les auteurs prévoient donc des abstractions, que nous qualifions d'intensionnelles, faisant passer d'un langage de haut niveau \mathcal{L} à un langage abstrait \mathcal{L}_A , ainsi que d'abstractions que nous qualifions d'extensionnelles. L'univers des *configurations* possibles dans la représentation perçue \mathcal{P} est noté Γ et une abstraction, faisant passer d'un cadre \mathcal{R} à un cadre abstrait \mathcal{R}_A , repose sur une fonction de Γ à l'univers des configurations abstraites Γ_A .

Abstractions intensionnelles

Une première remarque est que notre notion d'abstraction intensionnelle p_A est interne : elle est définie de \mathcal{L} vers \mathcal{L} , ce qui permet d'exprimer la contrainte caractéristique de notre notion d'abstraction : \mathcal{L}_A est fermé par l'opérateur de spécialisation minimale $\vee_{\mathcal{L}}$ de \mathcal{L} . De plus nous imposons l'idempotence (un élément abstrait par p_A ne peut être lui-même abstrait en un élément différent par p_A) et la préservation de l'ordre partiel : si $t \succeq t'$ alors $p_A(t) \succeq p_A(t')$. Les abstractions

4. Nous omettons ici l'indice g , pour grounded (ou instancié), de la notation originale.

préservant l'ordre partiel ont un rôle particulier en apprentissage [SZ98]. En particulier, lorsque il existe un énoncé maximale spécifiquement \dot{x} dont l'extension contient l'instance x , on a toujours : si $\dot{x} \succeq h$ alors $p_A(\dot{x}) \succeq p_A(h)$. On a les propriétés suivantes :

Proposition 15

1. Pour tout h de \mathcal{L} , $\{x\} \subseteq \text{ext}(h)$ ssi $h \preceq \dot{x}$
2. Si p_A est interne et idempotente, alors pour tout h' de $p_A(L)$, $\{x\} \subseteq \text{ext}(h') \Rightarrow h' \preceq p_A(\dot{x})$
3. Si de plus $\forall h, p_A(h) \preceq h$, c'est-à-dire p_A est une projection, alors :
Pour tout h' de $p_A(L)$, $\{x\} \subseteq \text{ext}(h')$ ssi $h' \preceq p_A(\dot{x})$

A. Giordana et collaborateurs [GSF⁺91, GRS91, GS90] définissent en particulier une *abstraction sémantique*, comme une définition de nouveaux concepts à l'aide d'une théorie qui satisfait la préservation de l'ordre. La théorie prend la forme d'une définition des prédicats de \mathcal{L}_A à partir de ceux de \mathcal{L} comme par exemple :

$$\text{BRICK}(x) \leftrightarrow (\text{leftmost}(x) \wedge \text{rightmost}(x) \wedge \text{on}(x, y)) \vee (\text{leftmost}(x) \wedge \text{rightmost}(x) \wedge \text{on}(y, x))$$

qui peut s'écrire aussi sous la forme de deux clauses⁵. Plus récemment, le programme INTHE-LEX a été conçu pour apprendre de telles abstractions à partir d'observations[FBDME05a]. Deux formes de règles sont construites : celles regroupant des littéraux de \mathcal{L} et les remplaçant par un seul littéral de \mathcal{L}' et celles éliminant un groupe de littéraux. Cette recherche d'un langage abstrait est à rapprocher des travaux sur les macro-opérateurs d'E. Alphonse[Alp04].

Dans un premier temps nous allons nous restreindre aux définitions n'impliquant qu'une clause et rester à l'ordre 0. Il est clair dans ce cas que le langage \mathcal{L}_A obtenu est isomorphe au sous-langage $A = p_A(L)$ obtenu en fermant par $\vee_{\mathcal{L}}$ l'ensemble des corps de définitions. Ici $\vee_{\mathcal{L}}$ est la conjonction logique \wedge :

Soit la théorie $\Theta = \{h_1 \leftrightarrow t_1, \dots, h_n \leftrightarrow t_n\}$ où les h_i sont des atomes de \mathcal{L}_A et les t_i des conjonctions d'atomes de \mathcal{L} . Soit

$$A = \{t \in \mathcal{L} \mid t = t_{i_1} \wedge \dots \wedge t_{i_k}\}$$

Il est clair que \mathcal{L}_A est fermé par conjonction logique et que tout élément $h_{i_1} \wedge \dots \wedge h_{i_k}$ de \mathcal{L}_A , est isomorphe à l'élément $t_{i_1} \wedge \dots \wedge t_{i_k}$ de A . D'autre part, A , en tant que sous-ensemble fermé par $\vee_{\mathcal{L}} = \wedge$ de \mathcal{L} est une abstraction et donc associé à une projection p_A . p_A préserve donc l'ordre et satisfait la propriété 3 de la proposition 15, essentielle pour l'apprentissage. Cette propriété est donc aussi satisfaite pour le langage \mathcal{L}_A .

Le passage de notre notion d'abstraction intensionnelle à un langage \mathcal{L} d'ordre 1 est envisageable, et peut-être simplifié par l'utilisation de macro-opérateurs, mais une discussion de ce sujet dépasse la cadre de ce mémoire. Enfin si l'on considère le cas général où le corps de la définition est disjonctif, en ordre 0, cela suppose d'avoir un langage \mathcal{L} permettant d'exprimer de tels termes, ce qui suppose de généraliser au delà de la simple disjonction logique. Des travaux récents [WL08, DS04] étendent l'analyse de concepts formels aux langages de formes normales

5. BRICK représente une unité de construction, qui est libre à gauche, libre à droite et au dessus, ou au dessous d'un objet. Il s'agit dans le travail évoqué [GSF⁺91] d'un changement de représentation profond : la brique p est obtenue en regroupant deux blocs h et m de la représentation fine, et les propriétés de p , comme $\text{rightmost}(p)$ sont dérivées des propriétés des blocs h et m .

disjonctives monotones, c'est-à-dire exhibent une correspondance de Galois entre intensions disjonctives et extensions. La notion d'abstraction proposée ici, basée sur un sous-ensemble fermé par $\vee_{\mathcal{L}} = \wedge$ de \mathcal{L} , s'étend naturellement à ces nouvelles correspondances.

Nous revenons maintenant sur la relation entre généralisation et abstraction à travers précisément une définition disjonctive exprimant que différentes formes d'objets représentées dans $\mathcal{L}_{\mathcal{S}}$ vont être abstraites dans \mathcal{L}_G en une forme plus générale :

$$QUADRILATERE(x) \leftrightarrow rectangle(x) \vee carre(x) \vee \dots$$

$$CURVILIGNE(x) \leftrightarrow cercle(x) \vee ellipseIrreguliere(x) \vee \dots$$

Ici, on remplace donc dans \mathcal{L}_A des formes particulières par des formes plus générales. Dans notre modèle, cette généralisation s'écrit comme la composition de deux opérations : de $\mathcal{L}_{\mathcal{S}}$ contenant uniquement les prédicats les plus spécifiques (*rectangle*, *carré*) on remonte (dans le treillis des abstractions) vers \mathcal{L} contenant à la fois les prédicats spécifiques et les prédicats généraux (*rectangle*, *carre*, *quadrilatère*), qui sera lui abstrait en \mathcal{L}_G ne contenant que les prédicats généraux. Ainsi, pour le même ensemble d'objets I , l'ensemble des concepts extensionnels augmente (par inclusion) lorsqu'on passe de $\mathcal{L}_{\mathcal{S}}$ vers \mathcal{L} et diminue dans l'abstraction de \mathcal{L} vers \mathcal{L}_G . La figure 3.5 représente les trois treillis de Galois pour \mathcal{L} , $\mathcal{L}_{\mathcal{S}}$ \mathcal{L}_G , et un ensemble de 3 objets : un rectangle bleu, un carré bleu et une ellipse bleue.

Considérons maintenant le cas d'objets représentés par leur couleur et une mesure t prenant ses valeurs dans $[1, 2, 3]$. Le langage \mathcal{L} contient les intervalles $t[1]$, $t[2]$, $t[3]$, $t[1, 2]$, $t[2, 3]$. Le langage abstrait $\mathcal{L}_{\mathcal{S}}$ ne contient que les intervalles $t[1]$, $t[2]$, $t[3]$. Ici une abstraction \mathcal{L}_G de \mathcal{L} est obtenue en considérant les intervalles $t[1, 2]$ et $t[2, 3]$, mais la fermeture par l'opérateur de spécialisation $\vee_{\mathcal{L}}$ conduit à considérer aussi l'intervalle $t[2]$. C'est exactement le langage \mathcal{L}_A obtenu avec la théorie Θ suivante :

$$T[1, 2](x) \leftrightarrow t[1](x) \vee t[2](x)$$

$$T[2, 3](x) \leftrightarrow t[2](x) \vee t[3](x)$$

On perd ici par fermeture, sauf pour les bornes $t = 1$ et $t = 2$, la généralisation en intervalles d'au moins 2 éléments de t . Il est bien sûr possible de considérer un langage \mathcal{L}_{G2} non fermé par $\vee_{\mathcal{L}}$. Cependant la description \dot{x} d'un objet x tel que $t = 2$ n'est alors plus unique, elle devient $\dot{x} = \{t[1, 2], t[2, 3]\}$. Dans ce cas, si la relation d'équivalence extensionnelle est tout de même plus fine (les extensions associées à \mathcal{L}_{G2} sont un sous-ensemble de celles associées à \mathcal{L}_G) on peut avoir plusieurs termes maximaux dans une classe d'équivalence. Par exemple, considérons $I = \{x\}$ avec $\dot{x} = \{t[1, 2], t[2, 3]\}$, on a alors deux classes d'équivalences extensionnelles correspondant aux extensions \emptyset , et $\{e\}$ mais deux termes maximaux dans \mathcal{L}_G , $t[1, 2]$ et $t[2, 3]$ d'extension $\{e\}$. Une possibilité est de multiplier chaque objet x en plusieurs objets virtuels chacun associé à une description unique (en les étiquetant de leur objet d'origine). On retrouvera alors une description unique pour chaque objet (virtuel) et un treillis d'intension/extension dont les intensions appartiennent à \mathcal{L}_{G2} . On peut alors chercher des règles d'associations ou utiliser l'apprentissage supervisé par exploration de l'espace des fermés comme cela a été présenté au chapitre précédent.

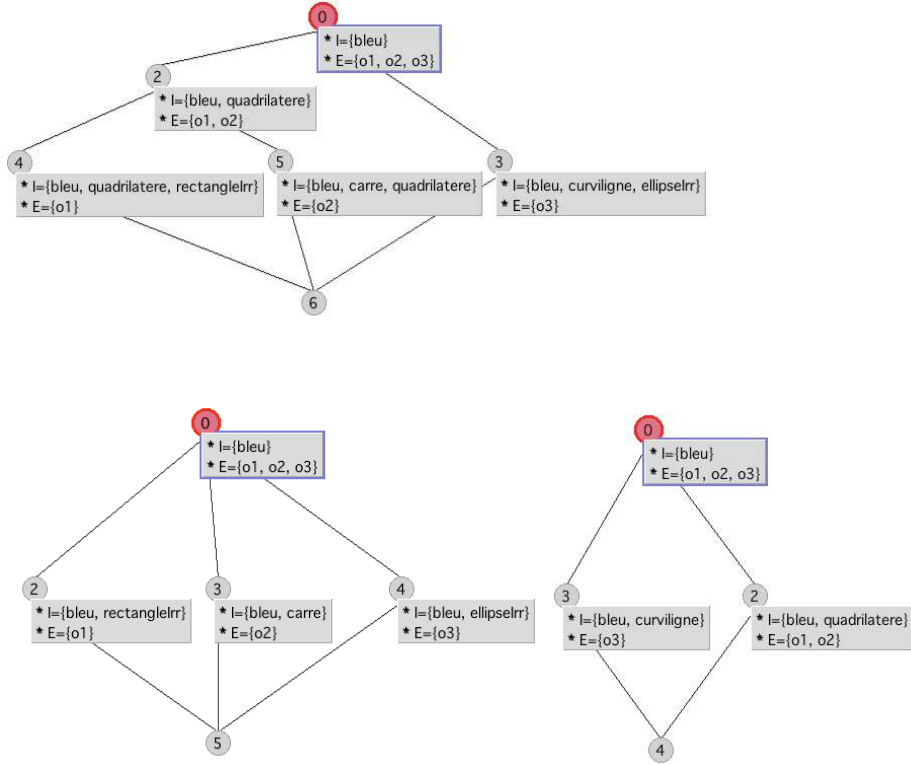


FIGURE 3.5 – Le treillis de Galois T_L , et les deux treillis abstraits intensionnellement T_{L_S} et T_{L_G} pour un rectangle bleu, un carré bleu et une ellipse bleue. T_L est en haut. En bas à gauche T_{L_S} est obtenu en ne gardant que les descripteurs de forme les plus spécifiques *carre*, *rectangle*, *cercle*, *ellipseIrreguliere*. En bas à droite T_{L_G} est obtenu en ne gardant que les descripteurs de forme les plus généraux, *quadrilatere* et *curviligne*. Les deux treillis T_{L_S} et T_{L_G} ne sont pas comparables selon l'ordre d'abstraction, mais sont plus abstraits que T_L .

Enfin, nous rencontrerons précisément cette situation dans le chapitre suivant concernant la recherche de mots répétés lorsqu'on généralise le mot, par exemple $A \times B \times C$, en motifs comme $\{A, B\} \times \{A, B\} \times \{B, C\}$ ⁶. Nous verrons que le choix qui est fait alors est de tenir compte de la multiplicité de la représentation d'un mot, ce qui amène à modifier sensiblement les algorithmes de recherche.

La figure 3.6 représente les trois treillis de Galois pour \mathcal{L} , \mathcal{L}_S et \mathcal{L}_G , et un ensemble de 3 objets bleus de taille respectives 1, 2 et 3. La figure 3.7 représente le treillis $T_{\mathcal{L}_{G^2}}$ obtenu en dupliquant les objets, de manière à leur assurer une description unique.

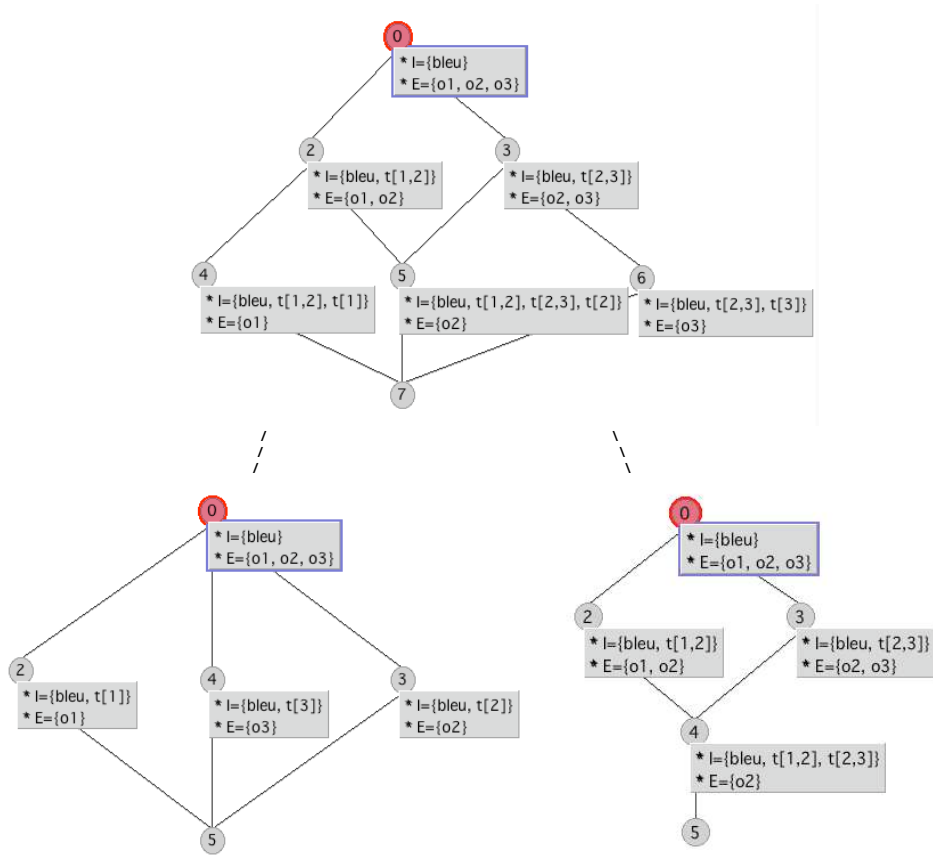


FIGURE 3.6 – Le treillis de Galois $T_{\mathcal{L}}$, et les deux treillis abstraits intensionnellement $T_{\mathcal{L}_S}$ et $T_{\mathcal{L}_G}$ pour trois objets bleus de taille 1, 2 et 3. $T_{\mathcal{L}}$ est en haut. En dessous à gauche $T_{\mathcal{L}_S}$ est obtenu en ne gardant que les descripteurs de taille $t[1], t[2], t[3]$. En dessous également de $T_{\mathcal{L}}$, $T_{\mathcal{L}_G}$ est obtenu en ne gardant que les descripteurs de taille les plus généraux $t[1,2], t[2,3]$.

Abstractions extensionnelles

Nous pouvons identifier notre univers d'instances \mathcal{X} à l'ensemble des configurations perçues Γ du modèle KRA.

6. A ou B en position 1, A ou B en position 2, B ou C en position 3

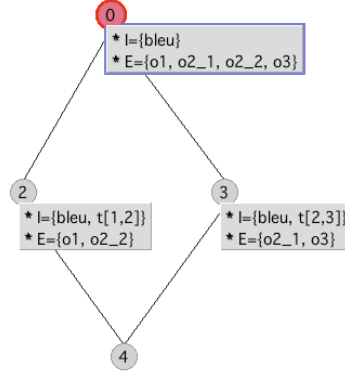


FIGURE 3.7 – Le treillis $T_{\mathcal{L}_{G_2}}$. $T_{\mathcal{L}_{G_2}}$ est obtenu en dupliquant les objets dont la description est multiple dans \mathcal{L}_{G_2} .

Considérons l'exemple suivant : Les objets sont des images de résolution 2×2 . Un élément γ de Γ est l'assignation d'un label M ou B à l'image et d'une intensité $x_{i,j}$ entre 0 et 10 à chaque pixel (i, j) . Le modèle KRA permet d'abstraire γ en une configuration γ_A associée à la même image perçue avec une résolution d'un seul pixel $(1, 1)$ et dont l'intensité est l'intensité moyenne sur les 4 pixels de l'image 2×2 . Ainsi le pixel "abstrait" se voit attribué une intensité moyenne $\bar{x} = (x_{1,1} + x_{1,2} + x_{2,1} + x_{2,2})/4$ ⁷. Dans notre cadre, nous pouvons rassembler en un groupe $C(\gamma_A)$ toutes les configurations identiques à γ sauf pour les intensités des pixels, mais dont le niveau d'intensité moyen \bar{x} est celui de γ . Nous partons donc de $\mathcal{P}(\Gamma)$ et notre abstraction est engendrée par le recouvrement, ici une partition, $\mathcal{C} = C_0, \dots, C_{10}$.

Remarquons que dans notre modèle nous ne transformons pas la représentation originale. Ainsi pour tout terme de t de \mathcal{L} , par exemple " $t = \text{intensité moyenne de l'image} > k$ ", on peut calculer s'il est satisfait par une image o donnée. Supposons que o soit représentée par la configuration $\gamma = (\text{label} = M, x_{1,1} = 2u, x_{1,2} = 0, x_{2,1} = 2u, x_{2,2} = 0)$ et donc abstraite dans KRA en $\gamma_A = (\text{label} = M, \bar{x} = u)$. t sera considérée comme vrai dans notre modèle si toutes les configurations du groupe auquel appartient γ , satisfont t , ce qui est le cas si et seulement si $\gamma \in \gamma_A$. Dans KRA comme dans notre modèle t est vrai ssi $u > k$. Dans KRA l'abstraction s'accompagne d'un changement de langage : un terme comme $t' = \text{l'intensité moyenne de la moitié supérieure de l'image est} > k$ n'est pas compatible avec l'abstraction réalisée au niveau des configurations. Dans notre modèle, t aura la valeur faux pour toute image o , car aucun groupe de configurations C_i ne renvoie vrai pour toutes ses configurations. Il est alors possible de ne considérer que les éléments d'extension abstraite non nulle de \mathcal{L} .

Plus profondément KRA est fondé sur les changements de représentation aux différents niveaux, une perception abstraite forçant un langage lui-même plus abstrait. Dans notre modèle il y a intrinsèquement un langage maximal, au sens de l'ordre sur les abstractions intensionnelles, c'est à dire de l'inclusion, \mathcal{L} , et une représentation maximale $\mathcal{P}(\mathcal{X})$ des ensembles d'extensions possibles des énoncés de \mathcal{L} . Ces deux représentations peuvent varier indépendamment. D'un point de vue extensionnel, l'idée de notre modèle est l'abstraction de l'individu vers le groupe, plutôt qu'un changement de perception. De plus notre modèle est basé sur la relation entre intension

⁷. ramené à un entier entre 0 et 10

et extension pour un ensemble donné d'instances $I \subseteq \mathcal{X}$, nous travaillons donc sur les concepts extensionnels distinguables sur cet ensemble particulier : du point de vue d'un mécanisme d'apprentissage seuls les concepts distinguables sur cet ensemble I sont intéressants, autrement dit les concepts intensionnels sont liés dans une relation d'équivalence plus grossière sur I que sur \mathcal{X} . En imposant un recouvrement $\mathcal{C} \subseteq \mathcal{P}(I)$ on obtient une relation encore plus grossière.

3.1.7 Granularité

On retrouve une notion de regroupement d'objets ou de modèles, dans différents travaux, en particulier ceux sur les *ensembles approximatifs* (rough sets [Paw96]). La notion d'ensemble approximatif est basée sur une relation d'*indiscernabilité* R entre instances d'un ensemble I . Dans le modèle initial cette relation est une relation d'équivalence, ce qui conduit à une partition de l'ensemble d'instances. Dans la mesure où les objets sont indiscernables, il est naturel de considérer qu'un sous-ensemble d'instances X à une borne inférieure \underline{X} , correspondant à l'ensemble des instances x telles que tous les objets indiscernables de x sont dans X , et une borne supérieure \overline{X} dont les instances sont indiscernables d'au moins une instance de X . De ce fait l'extension d'un terme, $ext(t)$, est comprise entre $\underline{ext(t)}$ et $\overline{ext(t)}$.

Les ensembles approximatifs ont été étendus aux relations non transitives [Yao98][Cat97][CC05]. Notre notion de granularité découle de l'extension des ensembles approximatifs aux ensembles approximatifs par recouvrement [Žak83] [BBW98] [Zhu09] [CC08]. Dans ce cas, les bornes supérieures et inférieures d'un sous-ensemble sont constitués de la réunion d'éléments du recouvrement [Zhu09]⁸.

La borne inférieure $\underline{X}_{\mathcal{C}}$ d'un ensemble X dans un *espace approximatif par recouvrement* (ou EAR) associé au recouvrement \mathcal{C} est exactement l'abstraction de X par \mathcal{C} , $p_{\mathcal{C}}(X)$. Un cas intéressant soulevé par exemple par G. Cattaneo [CC08] est celui des espaces *topologiques*, c'est à dire des abstractions fermées par intersection que nous avons présentées au paragraphe 3.1.1. Malheureusement, en ce qui concerne les *abstractions Alpha*, les recouvrements topologiques correspondants sont des partitions, et ne présentent donc pas d'intérêt.

Enfin un point important est l'interprétation *granulaire* de l'abstraction. En effet si on nomme *granules* les éléments \cup -irréductibles d'une abstraction extensionnelle A , nous avons vu que l'extension abstraite d'un terme t s'exprimait comme la réunion des granules inclus dans l'extension de t . Si on considère les granules comme des entités remplaçant les instances (voir les travaux de Y. Y. Yao [Yao04] dans le cas d'une partition) alors on a une nouvelle notion d'extension découlant de la définition 10. Des travaux récents portent sur les relations entre ensembles approximatifs par relation et approximations granulaires [MZLS07]. Nous nous en inspirons pour les définitions et notations ci-dessous⁹.

Définition 13 (Abstraction granulaire)

- $N(o)$ est le nom du granule o , où $o \in O_A$ est un élément \cup -irréductible de A . On note $N(O_A)$ l'ensemble des granules.
- $\underline{app}(e) = \{N(o) \in N(O_A) \mid o \subseteq e\}$ est l'abstraction granulaire de e .

8. Dans ce cas la borne supérieure peut être définie de plusieurs manières.

9. Cependant les travaux ne sont pas directement transposables. En effet le recouvrement considéré est constitué de voisinages de points de I , mais le lien de l'espace granulaire vers I est différent. Plus précisément, si on note $N(V(x))$ le granule associé à un voisinage $V(x)$, alors dans ces travaux $O(N(V(x_1)), \dots, N(V(x_k)))$ renvoie x_1, \dots, x_n alors que dans notre approche, $O(N(o_1), \dots, N(o_k))$ renvoie la réunion des granules $\bigcup \{o_1, \dots, o_k\}$.

- Soit n un élément de $N(O_A)$, $O(n) = \bigcup_{N(o) \in n} o$ lie l'ensemble des parties de $N(O_A)$ et l'abstraction A ¹⁰. $\underline{app}(a)$ est alors l'antécédent maximal de a par O .

Proposition 16 (Propriétés granulaires)

- $\underline{app}(e \cap f) = \underline{app}(e) \cap \underline{app}(f)$
- $\underline{app}(e \cup f) \supseteq \underline{app}(e) \cup \underline{app}(f)$
- $e \subseteq f \Rightarrow \underline{app}(e) \subseteq \underline{app}(f)$

Exemple Soit $O_A = \{\{a, b\}, \{b, c\}\}$. En utilisant la notation $N(\{a, b, \dots\}) = ab\dots$, on obtient :

$$\underline{app}(\{a, b\}) \cap \underline{app}(\{b, c\}) = \{ab\} \cap \{bc\} = \emptyset \text{ et}$$

$$\underline{app}(\{a, b\} \cap \{b, c\}) = \underline{app}(\{b\}) = \emptyset.$$

De même,

$$\underline{app}(\{a\}) \cup \underline{app}(\{b\}) = \emptyset \cap \emptyset = \emptyset \text{ et}$$

$$\underline{app}(\{a\} \cup \{b\}) = \underline{app}(\{a, b\}) = \{ab\} \supseteq \emptyset.$$

□

Nous allons définir le *treillis de Galois granulaire* liant le langage \mathcal{L} et l'ensemble $\mathcal{P}(N(O_A))$ des parties de l'ensemble de granules. Pour cela nous définissons la notion d'*extension granulaire* à partir de l'extension abstraite. De là nous tirons une notion d'*intension granulaire* en montrant que pour toute granule il y a un terme maximal dont l'extension contient le granule, et nous en déduisons la correspondance de Galois entre \mathcal{L} et l'ensemble $\mathcal{P}(N(O_A))$.

Définition 14 $ext_N(t) = \{N(o) \in N(O_A) \mid o \subseteq ext(t)\} = \underline{app}(ext(t)) = \underline{app}(ext_A(t))$, est l'extension granulaire de t .

Proposition 17 A tout granule $N(o)$ on peut associer un énoncé maximale spécifiquement $int_N(\{N(o)\}) = int(o)$ parmi tous les termes t tels que $N(o) \in ext_N(t)$, que l'on appellera description du granule.

Preuve (int, ext) formant une correspondance de Galois sur $(\mathcal{L}, \mathcal{P}(\mathcal{I}))$, $int(o)$ est l'énoncé le plus spécifique parmi tous les termes t de \mathcal{L} tels que $o \subseteq ext(t)$ (d'après la propriété 3 de la proposition 5), donc, d'après la définition 14, tels que $N(o) \in ext_N(t)$. □

Proposition 18

1. Pour toute partie $n = \{N(o_1), \dots, N(o_k)\}$, on a

$$int_N(n) = int(\{N(o_1)\}) \wedge_{\mathcal{L}} \dots \wedge_{\mathcal{L}} int(\{N(o_k)\})$$

2. (int_N, ext_N) forme une correspondance de Galois sur les treillis \mathcal{L} et $\mathcal{P}(N(O_A))$. On appellera treillis granulaire le treillis de Galois G_N correspondant.

3. $int_N(n_1 \cup n_2) = int_N(n_1) \wedge_{\mathcal{L}} int_N(n_2)$.

4. $ext_N(t_1 \vee_{\mathcal{L}} t_2) = ext_N(t_1) \cap ext_N(t_2)$.

10. Rappelons que tout élément de A a une unique décomposition maximale en éléments de O_A .

Preuve

1. Conséquence directe de la proposition 1, puisque la condition 1 de cette proposition est satisfaite d'après la proposition 17, et que la condition 2 de cette même proposition est satisfaite par hypothèse (\mathcal{L} est un treillis).
2. Conséquence de l'item précédent d'après la proposition 5.
3. Conséquence de la proposition 5.
4. Conséquence de la proposition 5.

□

Nous allons maintenant montrer que le treillis de Galois abstrait et le treillis de Galois granulaire sont isomorphes.

Proposition 19 *Les noeuds du treillis abstrait G_A et du treillis abstrait G_N sont en bijection en préservant les relations d'ordre entre les noeuds, et les intensions t_A et t_N des noeuds en correspondance sont égales.*

Preuve Il suffit de montrer que, pour un terme donné t , les fermés (la partie intensionnelle) coïncident, c'est à-dire $int_N \circ ext_N(t) = int \circ ext_A(t)$. Les noeuds ne peuvent alors être ordonnés que d'une seule manière :

$$int_N(ext_N(t)) = int_N(\underline{app}(ext_A(t))) = int_N(\{N(o) \mid o \subseteq ext_A(t)\}) = (\text{d'après la propriété 1 de la proposition 18}) \bigwedge_{o \subseteq ext_A(t)} int_N(\{N(o)\}) = \bigwedge_{o \subseteq ext_A(t)} int(o) = int(ext_A(t)) \quad \square$$

On en déduit que l'exploration par spécialisation et fermeture de \mathcal{L} est simplifiée du fait qu'on n'a besoin que de l'extension des éléments $\vee_{\mathcal{L}}$ -irréductibles de \mathcal{L} ¹¹.

Cependant cela n'est intéressant que si le nombre de granules $|O_A|$ est du même ordre ou plus petit que $|I|$. C'est le cas si O_A partitionne I , mais ce n'est pas le cas par exemple pour l'abstraction Alpha. En effet dans ce cas la projection p_{C_α} (voir Définition 8), et donc l'extension Alpha ext_{C_α} , ne nécessitent pas de parcourir toutes les parties de proportion α des éléments de \mathcal{C} , pour tester leur inclusion dans e (ou $ext(t)$), mais seulement de parcourir les éléments u_j de \mathcal{C} et de mesurer l'intersection de u_j avec e .

Exemple Supposons $\mathcal{C} = \{u_1, u_2\}$, $\alpha = 0.5$ et $|u_1| = |u_2| = 1000$. On a alors de l'ordre de \mathcal{C}_{50}^{100} granules dans O_{C_α} et seulement deux classes dans \mathcal{C} dont il faut considérer l'intersection avec e . □

Cependant les travaux sur les ensembles approximatifs et leur généralisations, granulaires ou non, pointent dans deux directions[Pol02]. D'une part un sous-ensemble X de I est considéré comme encadré par une borne inférieure, correspondant à notre notion d'abstraction, et une borne supérieure, or celle-ci ne correspondant pas à une projection, il n'y a pas d'encadrement d'un treillis de Galois par des treillis abstraits. Les travaux s'orientent alors vers l'étude des aspects algébriques ou logiques des encadrements des extensions par leurs bornes inférieures et supérieures. Notre point de vue sur l'abstraction consiste en une simplification de l'instance par l'ensemble des propriétés communes à sa (ses) catégorie(s) ce qui conduit à s'intéresser à la borne inférieure de l'encadrement. D'autre part, il y a un continuum entre ces deux bornes, une famille ordonnée par

11. voir propriété 4 de la proposition 18.

inclusion d'approximations de X basée sur la notion d'appartenance approximative. L'appartenance approximative $\mu_X(x)$ est un degré d'appartenance d'un élément x à un sous-ensemble X , tel que si on définit $x \in_\beta E$ ssi $\mu_E(x) \geq \beta$, les ensembles X_β obtenus varient de manière monotone de $X_1 = \underline{X}$ à $X_0 = \overline{X}$. Pour toutes les valeurs sauf $\beta = 1$, la fonction $ext_\beta(t) = ext(t)_\beta$ ne permet pas de former avec int une correspondance de Galois, donc de treillis d'intension/extension¹².

Les abstractions $A_\alpha = p_{C_\alpha}(\mathcal{P}(I))$ sont ordonnées : A_α est plus grossière que $A_{\alpha'}$ pour $\alpha \geq \alpha'$. Pour un sous-ensemble X de I donné, les ensembles $X_\alpha = p_{C_\alpha}(X)$ obtenus varient de manière monotone de $X_1 = \underline{X}$ à $X_0 = X$. Nous en parlerons plus en détail, ainsi que des treillis Alpha, des extensions Alpha et des implications Alpha, dans le paragraphe suivant.

Enfin il est intéressant de noter que Wolf Vanpaemel et Gert Storms ont proposé récemment en psychologie cognitive une notion d'abstraction proche de la nôtre, *the varying abstraction model of categorization* (VAM), pour la modélisation de ce qui est mémorisé lors de l'apprentissage d'une catégorie[VS08]. Dans ce cadre deux modèles sont classiquement opposés : celui basé sur un *prototype*, c'est-à-dire une représentation résumée des exemples perçus de la catégorie, et celui basé sur les *exemplaires*, c'est à dire sur la mémorisation de l'ensemble des exemples perçus, le premier ayant récemment décliné en faveur du second¹³. Dans leur modèle, les auteurs partent du modèle GCM à base d'exemplaires et de sa contrepartie MPM basé sur un prototype, et proposent le modèle VAM dans lequel les exemplaires sont regroupés (en formant une partition) pour former un ensemble de prototypes, la taille des groupes faisant varier le niveau d'abstraction. Dans ce modèle, chaque prototype est le centroïde d'un groupe (un exemplaire moyen). Dans notre approche le groupe est représenté par la description d'un granule.

3.2 La représentation Alpha : abstraction, extension, treillis Alpha des intensions/extensions

Le point de départ des travaux sur les représentation Alpha est un travail de N. Pernelle, V. Ventos et M-C. Rousset[PRV01]. Dans ce travail, les auteures utilisaient une partition a priori des instances en *classe de base* et deux niveaux de langages, d'expressivité croissante \mathcal{L}_2 et \mathcal{L}_3 . Le propos était de construire, je cite : "de façon automatique un ensemble de classes regroupant des données en fonction de l'existence de propriétés communes, en associant à chaque classe une description permettant de rendre compte de ces propriétés communes". J'ai rejoint ce travail pour l'étendre à la construction de treillis de Galois correspondant à des grains de représentations plus ou moins grossiers. Nous avons alors défini l'extension Alpha pour représenter une échelle d'abstraction reposant sur une partition a priori en classes. Nous avons également défini l'ordre partiel, par projections intensionnelles et extensionnelles, sur les treillis de Galois intension/extension et conçu le système interactif ZOOM dans le but de naviguer dans un treillis potentiellement de très grande taille, en passant de treillis grossiers, intensionnellement et extensionnellement, à des

12. Le problème ici est qu'en général $ext(int(ext_\beta(t))) \supset ext_\beta(t)$.

13. On retrouve là le débat classique en apprentissage supervisé opposant les méthodes représentant d'une manière ou d'une autre la classe à apprendre ou une frontière délimitant celle-ci, et les méthodes de type plus proches voisins. Récemment, en apprentissage symbolique supervisé, S. Salzberg[Sal91], P. Domingos [Dom96] ainsi que Pierre Brézellec et collaborateurs (voir chapitre précédent) ont proposé une approche unifiant l'apprentissage par règles et l'apprentissage par instances. Dans tous ces cas, un ensemble de généralisation maximales spécifiques (ou lgg) de sous-ensembles d'instances représente le concept ou une classe à apprendre.

raffinements locaux dans des sous-treillis plus fins [PRSV02]. Dans une thèse récente au LORIA [Mes09], Nizar Messai décrit un système de navigation pour la découverte de ressources, dans un treillis de concepts multivalués. La navigation y a un aspect dynamique associé à des variations dans le degré de granularité des treillis explorés. Ce travail, qui s'interprète avec les notions d'abstractions proposées ici, concrétise et prolonge la proposition de ZOOM en mettant au centre du système interactif la navigation par requêtes.

Après avoir étudié les projections intensionnelles et extensionnelles, nous nous sommes concentrés sur l'étude des treillis Alpha, de la α -extension [VSL04b] et des α -implications correspondantes [VSL04a]. Comme nous l'avons vu au paragraphe 3.1.1, l'extension ext_α se définit comme une abstraction, via une projection p_α , de l'extension classique : $ext_\alpha = p_\alpha \circ ext$, et le treillis de Galois abstrait correspondant G_α^C , où C est la partition en classes de base, est appelé treillis Alpha. Nous avons montré que lorsque α augmente, le treillis Alpha croît en abstraction : si $\alpha' > \alpha$ alors $G_{\alpha'}^C$ est plus grossier (plus abstrait) que G_α^C . Cela exprime le fait que l'ensemble des α' -extensions est inclus dans l'ensemble des α -extensions et que la relation d'équivalence $\equiv_{\alpha'}$ entre termes du langage \mathcal{L} est plus grossière : si $ext_\alpha(t) = ext_\alpha(t')$ alors $ext_{\alpha'}(t) = ext_{\alpha'}(t')$. Les α -implications, exprimant l'inclusion des α -extensions, sont intéressantes pour le raisonnement approximatif. Le paragraphe ci-dessous est extrait de [VSL04a] :

... supposons que nous séparions des espèces animales (chacune correspondant à une instance) en les classes de base suivantes : *mammifères*, *oiseaux*, *insectes*. Nous cherchons à extraire des règles générales de ces données. Une règle intuitive est par exemple : "un animal qui *vole* devrait avoir des *ailes*. Cette implication est vraie aussi bien pour les oiseaux (les oiseaux coureurs, comme l'autruche, ne contredisent pas la règle) que pour les insectes. Pour être universelle, la règle devrait aussi être vraie pour les mammifères, qui en général ne volent pas, mais est contredite par l'écureuil volant. L'approche Alpha ici permet de tirer parti de ce que peu de mammifères volent (en d'autres termes, la prémisse de la règle n'est pas fréquente dans la classe *mammifère* à laquelle appartient l'individu qui falsifie la règle). Utiliser une α -extension permet de retirer l'écureuil-volant de l'extension de la prémisse de la règle. Ici, une valeur faible de α est suffisante pour obtenir une règle d' α -implication (avec donc une confiance de 1) exprimant que les animaux volants ont des ailes. Bien sûr des valeurs plus élevées de α , (proches de 100) évitent aussi la falsification de cette règle. Cependant dans ce dernier cas, une règle d' α -implication exprime quelque chose de différent : elle ne s'applique à un individu que lorsque la prémisse de la règle est commune à la *plupart* des individus de la même classe de base. Dans notre exemple, seuls les *oiseaux* seraient ainsi concernés par une telle règle mais pas les *insectes*.

Dans ces premiers travaux le recouvrement \mathcal{C} utilisé (l'ensemble des classes de bases) est une partition. Par la suite nous avons généralisé l'approche à un recouvrement quelconque [VS05a]. Comme nous l'avons vu au paragraphe 3 l' α -extension associée à un recouvrement \mathcal{C} quelconque est une extension abstraite pour un recouvrement \mathcal{C}_α plus fin que \mathcal{C} ¹⁴. Cependant l'idée de cette généralisation était pragmatique : il s'agissait d'autoriser une même instance à appartenir à deux classes différentes, ce qui permet d'exprimer la multiplicité dans la catégorisation d'un objet,

14. Inversement toute extension abstraite correspond à une 1-extension pour le même recouvrement.

tout en gardant le bénéfice de l'approche Alpha dans l'abstraction. Considérons par exemple, un individu appartenant à la fois à la catégorie "Parent" et à la catégorie "Travailleur" et possédant la propriété *stress*. Il pourra alors se trouver dans l'extension $ext_{0,1}(stress)$ si soit au moins 10% des Parents sont stressés, soit au moins 10% des Travailleurs le sont. Cette propriété peut lui être niée en revanche si il est dans la catégorie "Rentier"¹⁵. Par ailleurs nous nous sommes intéressés [VS05a, VS05b] aux règles d'associations Alpha (voir 3.1.5 pour les définitions dans le cas plus général des règles d'association abstraites) et à la construction des bases de règles correspondantes. Cela nous a également amené à définir, pour pouvoir exprimer les supports minimum des règles, les *treillis Alpha fréquents* : il s'agit simplement de combiner la notion de fréquence locale aux classes (Alpha) à une notion de fréquence globale sur l'ensemble des individus. Concrètement cela revient simplement à ne pas considérer les noeuds dont l' α -extension est de cardinal inférieur à un support minimum¹⁶.

Nous avons également remarqué que le nombre d' α -implications valides augmente, pour une même recouvrement, lorsque α , c'est à dire la granularité, augmente : les α -implications valides à une granularité fine restent vraies à une granularité plus grossière, ce qui est une conséquence de notre définition de l'abstraction. Cependant, expérimentalement, le nombre d' α -implications dans la base de Guigues-Duquennes¹⁷ diminue. Nous avons alors montré que, paradoxalement, si on considère la base *min-max* [PTB⁺05] constituée de la réunion des ensembles de couples (générateurs, fermé) associé à chaque noeud, alors le nombre d'implications diminue lorsque la granularité augmente [VS05b]. Nous en concluons que l'effet de l'abstraction est un enrichissement et une simplification de la structure (représentée par l'ensemble des implications valides).

Nous nous sommes alors intéressés, suite à une suggestion de Nathalie Pernelle, à la construction incrémentale *par classe* des treillis Alpha (et Alpha fréquents). En effet, d'une part, nous avons remarqué que la suppression de classes dans un recouvrement (et donc des instances correspondantes¹⁸) est une abstraction. En particulier le treillis fréquent (pour un support α) de chaque classe est plus abstrait que le treillis α pour le recouvrement constitué de l'ensemble des classes. D'autre part, des travaux de P. Valtchev et collaborateurs existaient sur la fusion de treillis de Galois [VM01, VML02], correspondant, pour le cas particulier $\alpha = 0$, à la fusion des treillis Alpha des classes en le treillis Alpha du recouvrement. Nous avons alors étendu la méthode de P. Valtchev à la fusion de treillis abstraits (treillis projetés extensionnellement) et avons appliqué la méthode à la construction incrémentale *par classe*¹⁹ de treillis Alpha [VS05b].

Une implémentation de la construction incrémentale par classe des treillis Alpha et des règles d'associations correspondantes a été réalisée au LIPN sous la direction de Marc Champesme à partir de la plate-forme "Galicia"²⁰ consacrée aux treillis de Galois. Elle est opérationnelle²¹ et permet d'expérimenter l'ensemble des notions et méthodes présentées dans ce chapitre²².

15. pour les besoins de l'analyse et bien qu'il en éprouve les symptômes

16. Ils sont projetés, via une projection extensionnelle, sur le noeud \perp d'extension nulle.

17. précisément dans la base correspondant à celle de Guigues-Duquenne pour les α -extensions.

18. sauf si elles appartiennent également à une classe non supprimée

19. On ne peut en effet pas ajouter simplement des instances d'une classe existante mais seulement ajouter une nouvelle classe et ses instances.

20. <http://www.iro.umontreal.ca/~galicia>

21. <http://www-lipn.univ-paris13.fr/~champesme/alphabetagalicia>

22. Les recouvrements que l'on peut définir n'étant pas restreints aux partitions on peut expérimenter pour un langage d'attributs $\mathcal{P}(A)$ les treillis abstraits en toute généralité, en considérant le cas $\alpha = 0$, ou faire varier *alpha* pour étudier plus précisément les treillis et règles d'associations Alpha.

A titre d'exemple, considérons la table 3.2 dont les instances sont partitionnées en *Mammifères*, *Insectes* et *Oiseaux*.

Les treillis fréquents ($\alpha = 0.6$) des classes Insectes et Oiseaux construits par "Alphabetagalia"²³ sont représentés Figures 3.8 et 3.9.

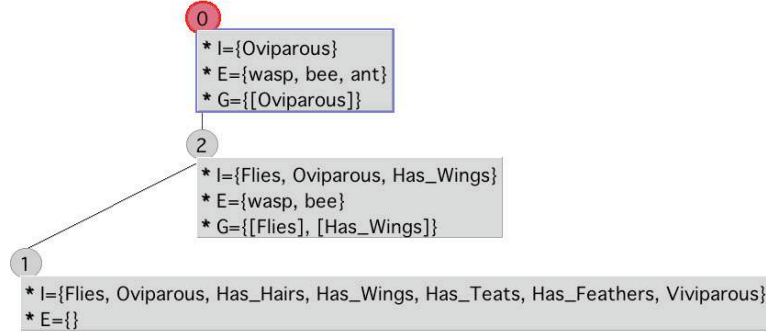


FIGURE 3.8 – Le treillis fréquent $G_{0.6}^{\{Insectes\}}$

Figure 3.10 est représenté le treillis Alpha ($\alpha = 0.6$) du recouvrement $\{Insectes, Oiseaux\}$ obtenu en fusionnant les treillis fréquents ($\alpha = 0.6$) des classes *Insectes* et *Oiseaux*.

3.3 Perspectives

Les travaux en cours ou en projet concernant les treillis Alpha et abstraits portent sur les aspects logiques (il y a-t-il une logique de l'abstraction, des connecteurs modaux Alpha ? quelles sont ses propriétés ?), les études expérimentales sur des données de grande taille (recherche de règles d'associations), le transfert de la notion de granularité régulière liées aux projections Alpha d'un point de vue extensionnel à une notion de granularité régulière sur les langages intensionnels, et enfin l'utilisation de l'extension abstraite en apprentissage supervisé pour réduire l'espace de

23. "beta" ici fait référence au seuil de fréquence globale.

Animals	Flies	Wings	Feath.	Hairs	Vivip.	Teats	Ovip.
cow	0	0	0	1	1	1	0
flying sq.	1	0	0	1	1	1	0
tiger	0	0	0	1	1	1	0
rabbit	0	0	0	1	1	1	0
platypus	0	0	0	1	0	1	1
bat	1	1	0	1	1	1	0
ant	0	0	0	0	0	0	1
bee	1	1	0	0	0	0	1
wasp	1	1	0	0	0	0	1
condor	1	1	1	0	0	0	1
duck	1	1	1	0	0	0	1
ostrich	0	1	1	0	0	0	1
pigeon	1	1	1	0	0	0	1

TABLE 3.1 – $Tab(i, j) = 1$ si le $j^{ème}$ attribut est vrai pour la $i^{ème}$ instance.

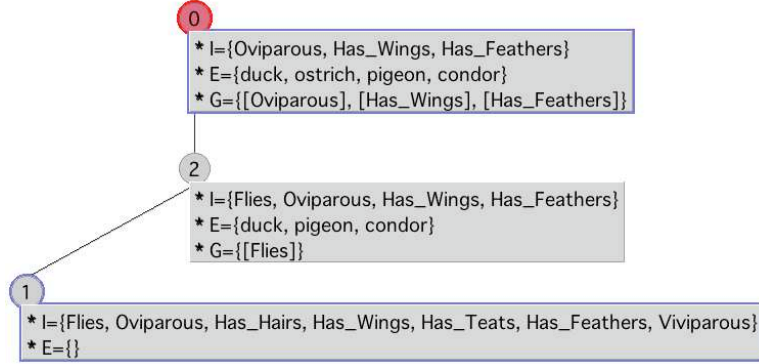
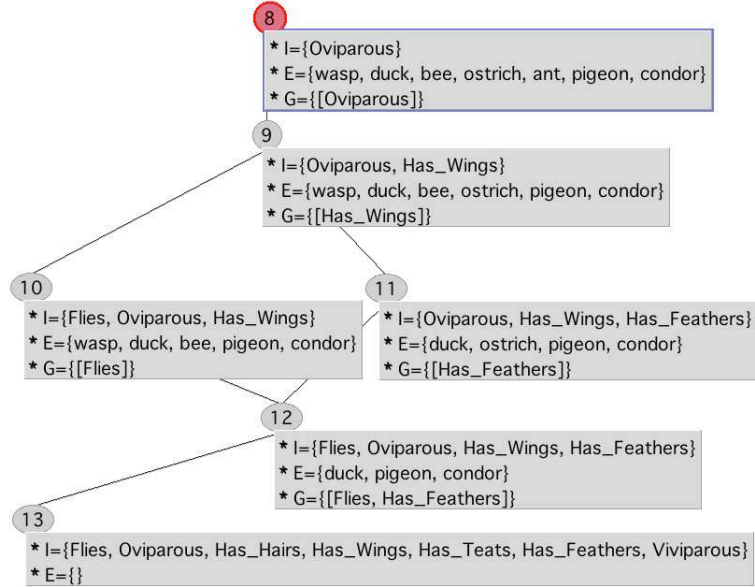
FIGURE 3.9 – Le treillis fréquent $G_{0.6}^{\{Oiseaux\}}$ 

FIGURE 3.10 – Le treillis Alphae $G_{0.6}^{\{Insects, Birds\}}$. Le terme fermé $\{Oviparous, Has_Wings\}$ (noeud 9) est le résultat de l'intersection des termes fermés $\{Flies, Oviparous, Has_Wings\}$ (noeud (2) de $G_{0.6}^{\{Insectes\}}$ Figure 3.8) et $\{Oviparous, Has_Wings, Has_Feathers\}$ (noeud (0) de $G_{0.6}^{\{Oiseaux\}}$ Figure 3.9).

recherche réel exploré, en réduisant le nombre de termes fermés.

Chapitre 4

Extraction de Motifs Séquentiels, Abstraits, Relationnels

4.1 Préambule

Ce chapitre décrit une série de travaux portant sur la recherche de motifs, trouvant leur origine dans la nécessité d’assouplir la recherche de motifs répétés dans les séquences biologiques, et menés à l’Atelier de BioInformatique (ABI), d’abord au sein de la section Physique-Chimie de l’institut Curie, puis au sein du département des sciences de la Nature et de la Vie de l’université Paris VI.

L’ABI est née au début des années 80 de la rencontre de chercheurs en sciences de l’Information et de chercheurs en Biologie, et a ainsi eu un rôle dans le développement de la recherche en BioInformatique en France (voir cet article d’Antoine Danchin [Dan00]). L’Atelier a gardé une structure informelle et a toujours pour vocation de servir de lieu de rencontre et d’échanges sur le traitement d’information en Sciences de la Vie. Depuis 1985 j’y mène une partie de ma recherche. Sur le thème général de l’analyse des séquences biologiques j’ ai participé à l’encadrement des thèses en Sciences de la Vie de Claudine Médique [Méd91] et d’Emmanuelle Ollivier [Oll90], ainsi qu’a celui des thèses en Informatique sur la recherche de motifs flexibles de Marie-France Sagot et Vincent Escalier, dont l’encadrant principal était Alain Viari. Marc Champesme, Dominique Bouthinon et Nalha El Zant dont j’ai encadré les thèses au LIPN, et Nadia Pisanti, en post-doc CNRS au LIPN y ont également travaillé sur l’extraction de motifs flexibles et leurs contributions présentées ci-dessous. Enfin il sera également question d’une partie du travail de thèse en Sciences de la Vie de Mathilde Carpentier, principalement encadrée par Joël Pothier ¹.

1. Les directeurs des différentes thèses mentionnées ci-dessus étaient respectivement Antoine Danchin, Alain Hénaut, Maxime Crochemore, Christian Choffrut, Daniel Kayser pour les 3 thèses LIPN, et Pierre Netter

4.2 Extraction de Motifs répétés sur des objets séquentiels et géométriques

4.2.1 Extraction de motifs dirigée par les données

Le problème abordé ici est celui de l'extraction de motifs fréquents, brièvement mentionné au paragraphe 2.3 dans le cadre général de la fouille de données, dans le cas particulier des objets séquentiels rencontrés en Analyse de séquences biologiques.

Les premiers travaux sur ce sujet visaient à chercher une méthode d'alignement par blocs d'un ensemble de protéines, associés à une même famille, c'est à dire partageant une même fonction biologique. Chaque protéine est représentée par une séquence d'acides aminés, c'est à dire techniquement par une séquence de symboles pris dans un alphabet de taille 20. La méthode de A.M. Landraud et collaborateurs dont nous sommes partis recherchait les mots les plus longs dont on trouvait des occurrences sur au moins k des n séquences [LAC89] et utilisait pour cela une adaptation de l'algorithme de recherche de mots répétés dû à Karp, Miller et Rosenberg (KMR) [KMR72]. Cependant un algorithme recherchant des répétitions exactes sur l'alphabet des acides aminés n'est pas approprié à la recherche de motifs répétés sur des séquences d'acides aminés. En effet un motif caractéristique dans une famille de protéines aura des occurrences similaires mais pas exactes, certains acides aminés ayant pu se substituer les uns aux autres, lors de l'évolution, sans que la fonction de la protéine ne soit perdue. Avec Alain Viari et Marc Champesme nous avons généralisé KMR à des alphabets *dégénérés* : dans les *mots* les symboles sont remplacés par des groupes de symboles, non disjoints², recouvrant l'alphabet. L'algorithme obtenu, nommé KMRC³[SVC95] considère une relation R non transitive entre symboles, les groupes de symboles étant les cliques maximales de cette relation.

KMR utilise un schéma de type *radix sorting* pour effectuer des intersections d'ensemble d'occurrences permettant ainsi la construction des extensions de mots longs à partir d'extensions de mots plus courts.

Plus précisément soit m et m' deux mots de longueur k , on notera $ext(m)$ l'ensemble des occurrences du mot m dans la chaîne S de longueur n ⁴, et $ext({}_k m') = ext(m') - k$ l'extension de m' lorsque l'on décale chaque occurrence de k vers la gauche. On a alors :

$$ext(m) \cap ext({}_k m') = ext(mm') \text{ avec } mm' = m +_k m' \quad (4.1)$$

$$ext(m) \cap ext({}_l m') = ext(m +_l m') \text{ avec un chevauchement sur } k - l \text{ positions} \quad (4.2)$$

m est ainsi le *préfixe* de longueur k de $m +_l m'$, et m' son *suffixe* de longueur k .

Exemple

$S = abbaababbaaa$, $n = 12$, $k = 2$, $ext(ab) = 1, 5, 7$, $ext(ba) = 3, 6, 9$, $ext({}_2 ba) = 1, 4, 7$, $abba = ab +_2 ba$ et $ext(abba) = 1, 5, 7 \cap 1, 4, 7 = 1, 7$.

En utilisant un chevauchement de $2 - 1 = 1$ positions on peut aussi construire $aba = ab +_1 ba$ et obtenir $ext(aba) = ext(ab) \cap ext({}_1 ba) = 1, 5, 7 \cap 2, 5, 8 = 5$ qui est bien la seule occurrence de aba dans S .

2. Dans le cas où les groupes sont disjoints, cela revient simplement à un changement d'alphabet.

3. KMR-Cliques

4. Une occurrence i est représentée par l'indice de début du mot dans la chaîne.

□

Ici l'exploration des motifs se fait par niveaux de taille croissante des mots et porte directement sur les extensions : les mots eux-mêmes n'interviennent pas, la méthode étant ainsi à la fois descendante et dirigée par les données, exploitant au mieux la structure séquentielle des données. A un niveau donné KMR partitionne les positions et la recherche des mots répétés de taille k dans une séquence est de complexité spatiale $O(n)$ et temporelle $O(n \log(k))$, la complexité temporelle de la recherche du mot répété le plus long étant $O(n \log(n))$. Cette faible complexité vient de ce que pour passer de la longueur k à la longueur $2k$ il suffit de parcourir les au plus n positions i des mots de taille k (les préfixes), rangées par extension (par mot), et d'aller directement dans la séquence en $i + k$ où on trouvera l'indice d'un mot de longueur k décalé de k (le suffixe) et associé à une entrée dans un vecteur. Chaque position d'un préfixe est donc envoyée à l'entrée du suffixe correspondant. A la fin de l'itération, un second parcours des indices permet de construire les extensions des mots de taille $2k$. Chaque étape nécessite donc $O(n)$ opérations et il y a au plus $O(\log(n))$ itérations, l'intersection de mots chevauchant permettant d'atteindre la taille k ou la taille k_{max} maximale d'un mot répété.

Des mots répétés aux motifs répétés : KMRC

L'efficacité de l'approche vient de cette exploitation de la structure de séquence pour distribuer les positions de la séquence et effectuer une intersection en $O(n)$. Nous avons autant que possible gardé cette notion de distribution, évitant en particulier d'engendrer des mots d'extensions vides, dans les différents développements de ce travail.

Une relation R non transitive entre symboles est typiquement obtenue en écrétant une matrice de similarité entre acides aminés (voir par exemple [LG08]). KMRC utilise le même schéma que KMR aux différences suivantes près :

- Une même position de l'extension d'un mot préfixe de longueur k peut être distribuée dans $O(g^k)$ indices de mots suffixes, où g est la dégénérescence de R , c'est-à-dire le nombre de groupes (cliques maximales de R) auquel appartient un symbole de l'alphabet. La conséquence est que, sauf dans le cas KMR, où $g = 1$, la complexité d'une itération va dépendre exponentiellement de la longueur du mot.
- On ne garde à la fin de chaque itération que les extensions *maximales* au sens de l'inclusion, ce qui nécessite un test d'inclusion qui peut être coûteux, et conduit à une complexité globale $O(n.k.g^{2k}.\log(k))$
- La méthode faisant une exploration exhaustive et en largeur peut conduire à une explosion en mémoire pour des longueurs relativement faibles.
- A chaque extension maximale, pour une valeur de k donnée, peuvent correspondre plusieurs motifs de même extension. L'extension peut alors être représentée par les groupes contraignant chaque position dans le motif.

Par la suite de nombreux développements ont été faits dans le cadre de la thèse de Marie-France Sagot[Sag96]. En particulier considérer directement un recouvrement de l'alphabet, plutôt qu'une relation, permet de représenter des motifs plus généraux, et ceci sans changer l'algorithmique de KMRC. La notion d'extension utilisée dans ce mémoire représente ici l'ensemble des occurrences d'un motif ou modèle dans une ou plusieurs séquences, c'est-à-dire un ensemble de positions

correspondant au début d'un mot qui est une *instance* d'un motif. La contrainte de fréquence ou répétition dans le cas d'une séquence, porte sur la taille de l'extension. Dans les problèmes posés en analyse de séquences, on est souvent amené à considérer une contrainte de *quorum* relativement à un certain nombre de séquences : c'est alors la présence d'une occurrence d'un motif dans au moins q séquences parmi n qui est recherchée. Cependant, dans la mesure où la contrainte portant sur l'extension est *anti-monotone* avec la taille du motif, c'est-à-dire : si mm' satisfait la contrainte alors m satisfait la contrainte, l'algorithmique de la recherche de motifs est inchangée. Nous donnons ci-dessous un exemple simple de motif répété trouvé par KMRC, lorsque l'on utilise un recouvrement.

Exemple Soient l'alphabet $A = \{a, b, c, d, e\}$, le recouvrement $\mathcal{C} = \{c_1 = \{a, b, c\}, c_2 = \{b, c, d\}, c_3 = \{d, e\}, c_4 = \{a, e\}\}$ et la séquence $S = adbeb$.

Ici la valeur maximale de g est 2 (par exemple a appartient à c_1 etc₄).

Pour $k = 3$ nous avons l'extension maximale $\{1, 3\}$ correspondant aux motifs $c_1 - c_3 - c_1$ et $c_1 - c_3 - c_2$ qui sont donc équivalents extensionnellement sur la séquence S . Nous noterons $c_1 - c_3 - (c_1, c_2)$ le motif extensionnel correspondant. Aux positions 1 et 3 on trouve les instances adb et beb du motif $c_1 - c_3 - c_1$.

$c_1 - c_3 - (c_1, c_2)$ est d'extension maximale mais pas $c_1 - c_2 - c_1$ dont l'extension $\{1\}$ est incluse dans celle de $c_1 - c_3 - (c_1, c_2)$.

Si nous considérons l'ensemble de séquences $\{S, S', S''\}$ avec $S' = adb$ et $S'' = eee$, alors $\{\{1, 3\}, \{1\}, \emptyset\}$ est une extension maximale, correspondant toujours aux motifs $c_1 - c_3 - c_1$ et $c_1 - c_3 - c_2$. Si $q=2$, alors cette extension satisfait la contrainte de quorum.

□

Lorsque KMRC doit construire les motifs de taille k , k n'étant pas une puissance de 2, il y a une itération où les mots préfixe et suffixes sont chevauchant et qui fait donc appel pour le calcul de l'extension à l'équation 4.2. Dans ce dernier cas, KMRC engendre des artefacts, appelés *pseudo-motifs* qui seront naturellement éliminés lors de la phase d'élimination des motifs non maximaux. En effet un pseudo-motif n'est pas constitué d'une classe d'équivalence de motifs de la forme $c_{i_1} - \dots - c_{i_n}$ mais contient, sur le chevauchement, des intersections de classes c_i . Ceci n'a qu'une incidence négligeable sur le coût d'exécution en temps et en espace de l'algorithme. Cependant ce surcoût devient problématique lorsque chaque itération concerne des mots chevauchants, avec $g > 1$, ce qui sera le cas lors de la recherche de motifs relationnels pour le problème de la recherche de motifs structuraux répétés (voir ci-dessous paragraphe 4.2.4). Il est heureusement possible d'éviter d'engendrer ces pseudo-motifs, comme nous allons le voir dans l'exemple ci-dessous :

Exemple

Soient l'alphabet $A = \{a, b, c, d, e\}$, le recouvrement $\mathcal{C} = \{C_1 = \{a\}, C_2 = \{b, c\}, C_3 = \{c, d\}, C_4 = \{e\}\}$ et la séquence $S = abeXXaceXXadeXX$ ⁵. En notant $\#i$ le $i^{\text{ème}}$ motif extensionnel, on a les motifs maximaux suivants :

5. Le symbole 'X', n'étant pas dans le recouvrement, n'apparaîtra pas dans les occurrences, et est utilisé ici par souci de lisibilité.

Identificateur	Extension	motifs associés
#1	1, 6	$C_1 - C_2$
#2	6, 11	$C_1 - C_3$
#3	2, 7	$C_2 - C_4$
#4	7, 12	$C_3 - C_4$

Si on construit les mots de taille 3, on utilisera deux mots préfixes et suffixes de taille 2, avec un chevauchement d'une lettre. On aura en particulier ici :

$ext(\#1) \cap ext(\#4) = \{6\}$ représentant le pseudo-motif de longueur 3 $C_1 - (C_2 \cap C_3) - C_4 = \{a\}\{c\}\{e\}$. Cet artefact sera éliminé par $ext(\#1) \cap ext(\#3) = \{1, 6\}$ représentant $C_1 - C_2 - C_4$ ou par $ext(\#2) \cap ext(\#4) = \{6, 11\}$ représentant $C_1 - C_3 - C_4$.

Il est clair ici que le problème est que lors de la génération d'un pseudo-motif, le suffixe du motif préfixe (C_2 de $C_1 - C_2$) ne correspond pas au préfixe du motif suffixe (C_3 de $C_3 - C_4$). Cependant si on transmet à l'étape $k = 3$ pour chaque motif maximal, de longueur 2, les identificateurs du motif préfixe et du motif suffixe (de longueurs 1) dont il résulte, alors une simple comparaison d'identificateurs permettra de savoir si deux motifs maximaux peuvent être combinés sans engendrer un artefact.

Considérons le motif #1 ($C_1 - C_2$) et les identificateurs de ses motifs préfixes et suffixes à l'étape $k = 1$ (notons les $p_{\#1}(C_1)$ et $s_{\#1}(C_2)$) ainsi que le motif #4 ($C_3 - C_4$) et les identificateurs préfixes et suffixes $p_{\#4}(C_3)$ et $s_{\#4}(C_4)$. Il suffit de remarquer que $s_{\#1}(C_2) \neq p_{\#4}(C_3)$.

Il faut de plus utiliser un mécanisme d'héritage des identificateurs préfixes et suffixes. En effet lorsqu'un mot est éliminé car non maximal, alors il faut que ses identificateurs préfixes et suffixes soient transmis au motif qui l'a éliminé. Il en résulte qu'un mot maximal sera associé à une liste de couples d'identificateurs. Voici un exemple montrant la nécessité de cet héritage :

Soient l'alphabet $A = \{a, b, c, d, e, f\}$, le recouvrement $\mathcal{C} = \{c_1 = \{a\}, c_2 = \{b, c\}, c_3 = \{c, d\}, c_4 = \{e\}, c_5 = \{f\}\}$ et la séquence $S = abf \dots ace \dots ade \dots fde$ ⁶. On a pour $k = 2$ les motifs maximaux suivants :

Identificateur	Extension	motifs associés
#1	1, 11, 21	$c_1 - c_2$
#2	11, 21	$c_1 - c_3$
#3	12, 22	$c_2 - c_4$
#4	12, 22, 32	$c_3 - c_4$

Après élimination des motifs non maximaux, il ne reste que #1($C_1 - C_2$) et #4($C_3 - C_4$) qui ne seront donc pas combinés, C_2 et C_3 (considérés comme des identificateurs) ne correspondant pas, ce qui empêchera d'obtenir le mot maximal d'extension $\{11, 21\}$ ($C_1 - (C_2, C_3) - C_4$). Cependant si on considère que #1 ayant éliminé #2 hérite de ses identificateurs, alors il est représenté par son extension ainsi que par ses listes d'identificateurs préfixes et suffixes $p_{\#1} = \{C_1\}$, $s_{\#1} = \{C_2, C_3\}$. Il en sera de même pour #4 qui hérite de #3 et est représenté par $p_{\#4} = \{C_3, C_4\}$, $s_{\#4} = \{C_4\}$. Il suffit alors de remarquer que $s_{\#1} = \{C_2, C_3\} \cap p_{\#4} = \{C_2, C_3\} \neq \emptyset$ pour autoriser la construction de $C_1 - (C_2, C_3) - C_4$ ⁷.

6. On ne note ici que les mots commençant aux positions 1 modulo 10.

7. Il faut bien distinguer ici le motif $C_1 - (C_2, C_3) - C_4$ qui représente une classe de motifs (par héritage des classes C_i) du pseudo-motif $C_1 - (C_2 \cap C_3) - C_4$. Dans le motif ci-dessus il faut comprendre (C_2, C_3) comme une réunion : l'un au moins des motifs $C_1 - C_2 - C_4$ ou $C_1 - C_3 - C_4$ a comme extension $\{11, 21\}$, l'autre étant d'extension incluse ou égale à $\{11, 21\}$. Dans le cas présent les deux ont la même extension, ce qui signifie qu'en position 2 on ne trouve que des symboles à l'intersection de c_2 et de c_3 .

□

4.2.2 Occurrences approchées de motifs

Dans sa thèse, Marie-France Sagot, a en particulier assoupli la notion d'occurrence de manière à inclure dans l'extension d'un motif toute position où l'on commence un mot dont la distance de Levenshtein à une instance du motif m est inférieure à un certain ϵ . Il s'agit en substance d'autoriser des erreurs dans l'instance d'un motif : le nombre d'opérations (substitution d'un symbole, insertion d'un symbole, suppression d'un symbole) pour passer du mot commençant en i à une instance du motif m est borné. Il s'agit ici de la manière classique de comparer deux (fragments de) séquences biologiques (ARN, ADN et protéines) en fonction d'évènements mutationnels ayant conduit, d'un ancêtre commun dans l'évolution, à ces deux séquences. Dans les algorithmes de recherche de motifs répétés, ainsi étendus aux occurrences d'erreur bornée, l'approche en largeur, qui conduit à des difficultés parfois insurmontables en espace mémoire, peut-être remplacée par une approche en profondeur [SVS97]. Ainsi l'algorithme LePoivre permet la recherche de motifs répétés avec une complexité en mémoire $O(n.k)$. Le prix à payer pour cela est faible en ce qui concerne le nombre d'itérations nécessaires : le facteur $\log(k)$ de l'exploration en largeur correspondant au nombre d'itérations pour construire les motifs de longueur k , passant à k . On perd cependant également la sélection d'extensions maximales (on ne peut plus comparer aisément les extensions des motifs) qui joue un rôle important lorsque le recouvrement est élevé ($g > 1$), ce qui est le cas dans les problèmes de recherche de motifs structuraux dans les protéines, mais pas dans les problèmes de recherche de motifs dans l'ADN. Dans ce dernier cas une version simplifiée de l'algorithme en profondeur partant directement de KMR est préférable [SEVS95].

4.2.3 Des motifs aux motifs relationnels

Enfin, nous avons étendu la notion de motif aux motifs relationnels. Pour cela nous considérons une séquence d'entités, représentées chacune un symbole, ainsi qu'un ensemble de relations entre les paires d'entités. On ajoute à l'alphabet de symboles et au recouvrement sur les symboles, un alphabet de relations ainsi qu'un recouvrement sur cet alphabet. Un k -motif séquentiel sera alors constitué de k groupes de symboles et de $k(k-1)$ groupes de relations. Ci dessous, les définitions formelles :

Définition 15 (Séquence relationnelle) Soit $R = \{r_1, \dots, r_{|R|}\}$ un alphabet de relations et $r \in R$ une relation. La séquence relationnelle t est une séquence de taille n sur l'alphabet A dans laquelle à chaque paire de positions (p_1, p_2) , telle que $1 \leq p_1 < p_2 \leq n$ et $|p_1 - p_2| \leq k-1$, on associe une unique relation $r \in R$ entre les positions p_1 et p_2 . Ce que nous écrivons $r(p_1, p_2)$ ou $(p_1, p_2) \in r$.

La notion de recouvrement de relation est définie ci-dessous :

Définition 16 Soit $G_R = \{CR_1, \dots, CR_{|G_R|}\}$, avec $CR_i \subseteq R$ pour $1 \leq i \leq |G_R|$, un recouvrement de relations sur R dont les CR_i sont appelés groupes de relations.

La notion de motif s'enrichit alors de contraintes relationnelles, et induit une notion d'occurrence plus contrainte :

Définition 17 Un k -motif relationnel est un k -motif plus les groupes de relations $CR_{(i,j)}$, avec $1 \leq i < j \leq k$, pour chaque paire de positions entre 1 et k . Un k -motif relationnel x a une occurrence dans t à la position p si le motif x a une occurrence à la position p de t et si pour toute les paires (i, j) on a $r(p+i, p+j)$ avec $r \in CR_{(i,j)}$.

Exemple Soient les recouvrements $\mathcal{C} = \{C_1 = \{a\}, C_2 = \{b, c\}\}$ et $\mathcal{G}_{\mathcal{R}} = \{CR_1 = \{r_a, r_b\}, CR_2 = \{r_b, r_c\}\}$ un motif relationnel M se représentera par exemple de la manière suivante :

$$M = \begin{array}{ccc} & CR_1 & \\ & CR_2 & CR_2 \\ C_1 & C_2 & C_1 \end{array}$$

Ce qui signifie que les symboles aux première, seconde et troisième positions doivent appartenir respectivement aux groupes C_1, C_2, C_3 , que la relation liant la paire de positions (1,2) appartient à CR_2 , celle liant la paire (1,3) appartient à CR_1 et celle liant la paire (2,3) à CR_2 . \square

La relation de récurrence pour calculer l'extension d'un $k+1$ -motif relationnel est basée sur la remarque suivante : à partir de deux k -motifs relationnels m et m' on obtient plusieurs $k+1$ -motifs relationnels selon le groupe de relation $CR_{(1,k+1)}$ contraignant la paire de position $1, k+1$, seule paire liée ni dans m ni dans m' . On notera $m +_1 m' + CR$ un tel $k+1$ -motif relationnel. La relation de récurrence permettant de construire les $k+1$ motifs à partir de deux k -motifs est alors la suivante :

$$ext(m) \cap ext({}_1m') \cap ext(CR_{(1,k+1)} = CR) = ext(m +_1 m' + CR) \quad (4.3)$$

Comme dans KMR et KMRC, seules les extensions non vides seront ainsi engendrées. Nous rechercherons des motifs et motifs relationnels répétés dans plusieurs problèmes biologiques mentionnés ci-dessous.

4.2.4 La recherche de motifs structuraux dans les protéines

Extraction de motifs structuraux séquentiels répétés : GOK

Nous avons utilisé KMRC pour chercher des motifs structuraux séquentiels répétés dans les squelettes $C - \alpha$ des protéines [SVPS95]. En effet la séquence d'acides aminés d'une protéine se replie dans l'espace en une conformation stable. Les acides aminés sont des molécules organiques se distinguant par leur chaîne latérale, autour d'un squelette identique contenant un atome de Carbone, dit Carbone- α . La succession des Carbone- α constitue le squelette de la structure, et constitue une chaîne dans l'espace dont la forme peut-être décrite par la succession de paires d'angles (ϕ, ψ) chacun associé à un acide aminé. Chaque fragment est ainsi décrit par des *coordonnées internes* : pour comparer deux fragments, il suffit de comparer leurs coordonnées internes⁸. La méthode a été intégré au logiciel interactif GOK, développé par Joël Pothier, qui a par la suite été

8. Cette approche s'oppose à la comparaison par *superposition* des fragments, dans laquelle la distance entre fragments passe par une superposition ce qui rend problématique les comparaisons multiples.

utilisé pour aligner avec succès une protéine de la famille des cytochromes P450 avec des protéines de la même famille dont les repliements étaient connus [JPD⁺97].

Extraction de motifs structuraux géométriques répétés : MESCOAN

A la même époque, dans le cadre de la thèse de Vincent Escalier, également encadrée par Alain Viari, nous nous sommes intéressés à l'extraction de motifs non plus séquentiels mais liés à la proximité dans l'espace des atomes, ce qui permet d'utiliser les chaînes latérales des acides aminés et est ainsi utile par exemple pour extraire certains motifs fonctionnels. Dans ce travail, concrétisé par le logiciel MESCOAN dédié à la recherche de motifs répétés dans plusieurs structures, l'occurrence d'un motif est un ensemble de points (d'atomes) dans l'espace décrit là aussi par des *coordonnées internes*, mais il s'agit cette fois de l'ensemble des distances inter-atomiques [EPSV98]. Nous avons gardé pour cela l'idée de construire les occurrences de motifs à partir de celles de motifs plus petits, mais la perte de l'ordre séquentiel a plusieurs conséquences :

- Les occurrences de motifs de taille k sont obtenues à partir de celles de deux motifs "frères" de taille $k - 1$ et partageant $k - 2$ atomes. Le motif étant relationnel, il faut également tenir compte de la distance entre les deux atomes n'appartenant pas à l'intersection des 2 motifs.
- Une des protéines joue un rôle particulier de *référence* : un motif sera un sous-ensemble d'atomes de la protéine de référence.
- Un ensemble d'atomes a_1, \dots, a_k appartenant à une protéine p est une occurrence d'un motif r_1, \dots, r_k extrait de la protéine de référence p_r si et seulement toutes les distances interatomiques sont similaires, c'est-à-dire si pour toute paire d'indices i, j , on $|dist(a_i, a_j) - dist(r_i, r_j)| \leq \epsilon$.
- On utilise également une contrainte de compacité : dans un motif, ou une occurrence, deux atomes ne peuvent être à une distance supérieure à une limite δ .

Extraction de motifs structuraux relationnels et séquentiels répétés : TRIADES

Cette description en coordonnées internes exprimant des relations entre entités (atomes) est à l'origine de la définition, dans le cadre de la thèse de Nalha El Zant [EZ05], de la notion de *motifs relationnels* pour lesquels un premier algorithme KMRRelat a été proposé [EZO4].

Nous avons alors appliqué KMRRelat à l'extraction de motifs structuraux séquentiels répétés, après avoir observé que dans le cas de GOK il arrivait parfois que deux fragments soient des occurrences d'un même motif, mais se superposent mal en raison d'un effet de *divergence*, c'est-à-dire d'une accumulation d'erreurs angulaires le long du motif. Cet effet est intrinsèque à toute méthode considérant séparément les éléments de la séquence. La première version du programme TRIADES développé par Joël Pothier et Mathilde Carpentier souffrait cependant d'explosion combinatoire au delà de quelques structures. Pendant le post-doc de Nadia Pisanti, nous avons proposé une amélioration de l'algorithme permettant d'éviter la génération de (parfois très nombreux) candidats inutiles lors de la recherche de répétitions. Le nouvel algorithme, KMROverlap, dont le principe a été décrit au paragraphe 4.2.1 a été implémenté au sein de TRIADES [PSC05, PSCP06][PSCP09].

4.2.5 Extraction d'une structure secondaire commune à une famille de fragments d'ARN

La notion de motif relationnel a été d'abord définie dans un cas particulier pendant la thèse de Dominique Bouthinon[Bou96]. Il s'agissait de rechercher une structure secondaire partielle commune à un ensemble de fragments d'ARN. L'ARN est constitué d'une séquence de nucléotides qui s'apparient selon certaines règles simples et de manière à minimiser l'énergie de la structure résultante. La structure secondaire est la représentation de cet ensemble d'appariements. L'observation de la structure étant difficile, des programmes de minimisation prédisent la structure d'un fragment d'ARN, avec cependant une certaine ambiguïté : parmi les n meilleures structures secondaires proposées, on trouvera souvent la structure correcte. La recherche d'une sous-structure commune est donc délicate pour deux raisons :

- La structure sera représentée par un ensemble de relations entre unités d'appariement appelées *hélices*, ou structures *tige-boucle* et constituées chacune d'une *tige* représentant l'appariement d'une suite de nucléotides avec une autre suite de nucléotides et d'une *boucle* représentant la partie non appariée entre les deux parties de la tige. Lorsque les hélices sont ordonnées selon leur ordre de départ dans la structure, il y a trois types de relations possibles entre deux hélices : l'*inclusion*, la *disjonction* et le *chevauchement*. Les motifs recherchés sont donc de nature relationnelle.
- La détection des unités potentielles est assez simple, mais la sélection des hélices se formant réellement dans la structure détermine la structure secondaire. Chaque structure candidate, parmi les n meilleures, est ainsi représentée par un sous-ensemble maximal d'hélices relativement à une relation de compatibilité entre 2 hélices. La structure d'une séquence d'ARN est ainsi représentée par un ensemble, (un *sac*) de structures *possibles*. La notion d'occurrence d'une sous-structure, c'est-à-dire un motif relationnel, dans une séquence, est alors définie comme l'occurrence du motif dans *une au moins* des structures possibles prédites.

Le problème de la recherche d'une structure commune à un ensemble de séquences, se ramène donc à la recherche de mots relationnels (il n'y a pas ici de notion de recouvrement sur les alphabets) répétés au sens d'une notion d'occurrence *crédule*. L'algorithme de recherche de mots relationnels répétés proposé par Dominique utilise l'ordre des hélices sur la séquence et la nature des relations pour coder progressivement les sous-structures dans un dictionnaire, ainsi qu'une relation de récurrence, adaptée au cas du dictionnaire, pour construire les extensions des mots répétés.

Un autre point intéressant de ce travail est l'étape d'ordonnancement des structures communes proposées par l'algorithme de recherche de répétitions. En effet, dans les expériences effectuées pour redécouvrir la structure en *feuille de trèfle* des ARN de transfert (une hélice contenant trois autres hélices disjointes entre elles) la structure correcte n'était pas distinguable, ni par sa taille, ni par sa fréquence, des autres structures candidates. La raison est que de nombreuses structures à base de chevauchement sont trouvées simplement parce qu'elles sont naturellement plus fréquentes. Faute de modèle probabiliste les fréquences de toutes les structures retenues ont été estimées dans des séquences aléatoires. La structure en feuille de trèfle, comme les structures déjà connues en général, ayant une probabilité *a priori* faible se retrouve alors en tête lorsqu'on corrige sa fréquence observée[BS99]. Enfin, notons l'intérêt constant pour la représentation proposée ici des structures

secondaires d'ARN en ensemble de 2-intervalles compatibles (selon la terminologie employée par Stéphane Vialette[Via04]) et spécifiquement pour la recherche des sous-ensembles maximaux de 2-intervalles compatibles [BFV07, CHL⁺08]. La figure 4.1, tirée de l'article de Stéphane Vialette précité, montre une structure secondaire d'ARN et sa représentation en 2-intervalles.

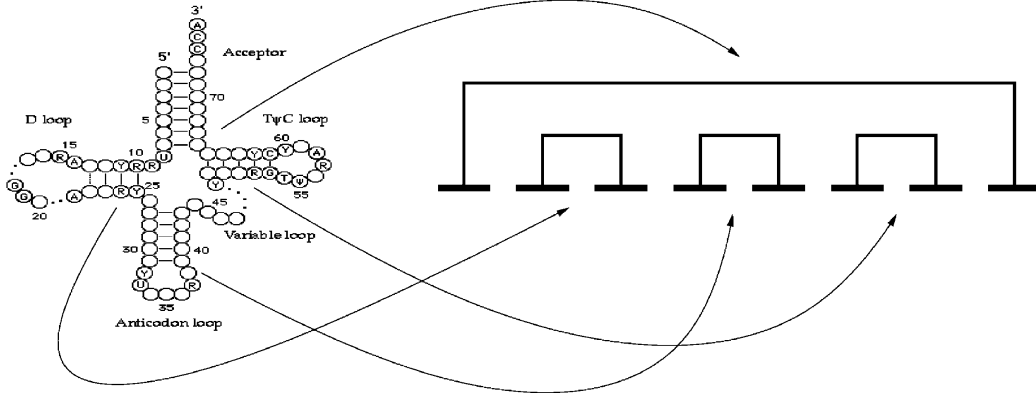


FIGURE 4.1 – La structure secondaire de l'ARN^t et sa représentation en un motif de 2-intervalles. Celle-ci s'écrit, en notant les 2-intervalles p_1, p_2, p_3, p_4 , ordonnés à partir de leur premier intervalle de gauche à droite : $inclus(p_1, p_2), inclus(p_1, p_3), inclus(p_1, p_4), disjoint(p_2, p_3), disjoint(p_2, p_4), disjoint(p_3, p_4)$.

Nous retrouverons ce travail dans le chapitre suivant, dans la mesure où, du point de vue de l'apprentissage artificiel et de la programmation logique inductive il permet de définir un nouveau type de problèmes d'apprentissage supervisé, les problèmes d'apprentissages à partir d'exemples ambigus, dont la formulation est proche mais différente de celle de l'apprentissage multi-instance [DLLP97] et qui généralise les problèmes d'apprentissage à partir d'exemples incomplets [BS98a].

4.3 Perspectives

Les différents algorithmes de recherche de motifs répétés sont des cas particuliers, séquentiels, de recherche de termes fréquents, tels que ceux évoqués dans les chapitres précédents. D'une part leur domaine d'application va au delà des problèmes de Biologie qui les ont inspirés, et en particulier il serait intéressant de les appliquer à des problèmes de fouille de séries chronologiques. Je pense en particulier à l'extraction de chroniques [DD99, Dou02], pour lesquels les motifs relationnels évoqués au paragraphe précédent sont parfaitement adaptés dans la mesure où les relations inclusion, disjonction, chevauchement sont un sous-ensemble des relations de l'algèbre d'intervalles de Allen [All83]. D'autre part, il serait intéressant d'étendre les motifs séquentiels relationnels proposés ici à des structures d'arbres et de graphes en essayant de garder l'approche purement extensionnelle et par niveau proposée ici. Cela pourrait être facilité par une meilleure compréhension des liens entre recherche de motifs séquentiels fermés ou *I*-maximaux (au sens d'une relation de spécificité entre motifs) [AU07] et recherche de motifs au sens extensionnel utilisé dans ce paragraphe.

Chapitre 5

Paradigmes

5.1 Apprentissage à partir de données ambiguës

Dans le problème d'apprentissage de structures secondaires d'ARN évoqué au paragraphe 4.2.5 les exemples présentent une forme extrême et originale d'incomplétude : la description d'une structure étant indirecte, aucun des éléments de description d'un exemple n'est connu, et il faut passer par une étape de prédiction des structures les plus probables, de laquelle on tire un ensemble de structures candidates. Chaque exemple est alors représenté par un ensemble de structures possibles. Ceci nous a amené, avec Dominique Bouthinon et Véronique Ventos à étudier les conséquences de situations d'ambiguïté ou d'incomplétude dans la représentation des exemples, sur l'apprentissage de concept, et plus précisément sur les cadres logiques d'apprentissages[De 97].

L'apprentissage de concept à partir d'exemples repose classiquement sur une relation de couverture entre une hypothèse h , prise dans un langage \mathcal{L} , et une instance e : $couvre(h, e^+)$ doit être vraie pour tous les exemples positifs e^+ et $couvre(h, e^-)$ doit être faux pour tous les exemples négatifs e^- , d'un ensemble d'apprentissage $E = E^+ \cup E^-$ représentant un certain concept-cible. h appartient alors à l'espace des solutions, appelé également *Espace des Versions* [Mit82] $V(\mathcal{L}, E)$ du problème lorsque le langage \mathcal{L} des solutions candidates est partiellement ordonné. Cette définition suppose une description *complète* de chaque exemple : pour toute hypothèse h et tout exemple e , $cov(h, e)$ doit être mesurable. Haym Hirsh a proposé d'étendre informellement cette notion de solution de manière à prendre en compte tout élément d'information, concernant les exemples, susceptible de contraindre l'Espace des Versions [Hir94]. La définition d'un problème d'apprentissage de concept doit alors également être révisée : une hypothèse h doit alors être *compatible* avec toute information, aussi partielle soit-elle, concernant un exemple.

Dans son travail sur les cadres logiques pour l'apprentissage de concepts, Luc De Raedt propose un cadre général, l'*apprentissage par satisfiabilité*, qui permet de rendre compte de l'ensemble des cadres d'apprentissage définis en Programmation Logique Inductive (PLI), en utilisant une notion de *réduction* d'un cadre d'apprentissage à un autre : tous les cadres explorés dans ce travail se réduisent à l'apprentissage par satisfiabilité, l'inverse n'étant pas vrai. Cependant l'apprentissage par satisfiabilité ne rend que partiellement compte de l'incomplétude des exemples. Sa limite d'expressivité se manifeste dans le traitement des exemples négatifs et a comme origine le maintien de la compatibilité d'une hypothèse avec un exemple négatif exprimée comme la négation d'une

relation de couverture. Nous avons proposé un cadre d'apprentissage, en situation d'incomplétude des exemples, en tirant les conséquences de l'approche de Haym Hirsh :

- La relation de compatibilité dépend de la nature de l'information fournie sur une observation et du label associé à celle-ci.
- Une hypothèse reste dans l'Espace des Versions lorsque l'information disponible sur un exemple ne permet pas d'éliminer cette hypothèse.

Il s'agit donc d'une approche *crédule*, dans l'esprit de l'Espace des Versions : le problème d'apprentissage se découpe, pour tout état d'information, en deux aspects :

- On définit, d'une part, l'ensemble des hypothèses candidates étant donné les informations disponibles.
- On applique d'autre part, une stratégie de préférence, pour retenir une hypothèse considérée comme solution courante.

Le cadre logique ne traite que de la première partie du problème, et le fait qu'elle soit *crédule* vise à garantir la convergence de l'apprentissage : de manière informelle, si aucune hypothèse n'est rejetée par erreur, alors, s'il existe une solution dans \mathcal{L} , et si toute information susceptible de contraindre l'Espace des Versions a une probabilité non nulle d'être rencontrée, la solution correcte finira par être trouvée, quelle que soit la stratégie de préférence utilisée. De ce point de vue, il s'agit de maintenir un Espace des Versions *correct*.

Dans le cas d'un exemple ambigu e associé à un label (+ ou -), représentant ce que l'on sait d'une certaine *observation* et représenté par un ensemble d'exemples complets possibles e_1, \dots, e_m , que nous appellerons simplement des *possibilités*, la relation de compatibilité exprime le fait que h doit être compatible avec au moins une possibilité de e . Ce qui conduit aux deux relations suivantes :

- Si e est positif, h est compatible avec e s'il existe $e_i \in e$, t.q. $couv(h, e_i)$, ce que nous écrirons $comp^+(h, e)$.
- Si e est négatif, h est compatible avec e s'il existe $e_i \in e$, t.q. $couv(h, e_i)$ est faux, ce qui est différent de la négation de $comp^+(h, e)$, et que nous écrirons $comp^-(h, e)$.

Nous avons proposé un cadre général pour l'apprentissage à partir d'exemples incomplets [BSV99], où l'incomplétude a deux sources différentes pour un exemple donné : certaines informations *utiles* à l'apprentissage sont inconnues, c'est la dimension *incertaine* ou ambiguë de l'incomplétude, et d'autres, connues ou non, sont considérées comme *superflues* (au sens employé par exemple dans [FBDME05b] ou [ZM09]) et conduisent à une abstraction de l'exemple, ce que nous appellerons la dimension de *généralisation* de l'incomplétude. On garde alors la notion de possibilité utilisée pour exprimer l'ambiguïté, mais une possibilité n'est plus un *exemple complet possible* mais une abstraction de celui-ci. Nous avons distingué dans ce travail deux formulations, étendant à l'incomplétude les principaux cadres logiques d'apprentissage :

- Une extension de l'apprentissage par interprétation, proposé par Luc De Raedt, dans lequel un exemple (complet) est représenté par une interprétation sur une base de Herbrand, et où la relation de couverture est : $couvre(h, e)$ ssi e est un modèle de h .
- Une extension de l'apprentissage par dérivation, le cadre dit *normal* de la Programmation Logique Inductive (PLI), dans lequel la description de chaque exemple est incluse dans une *théorie du domaine* B , et son label prend la forme d'un littéral instancié. On a alors $couvre(h, e)$ ssi $B, h \models e$.

L'apprentissage par dérivation avait été auparavant étendu en programmation logique, dans le cadre de l'apprentissage abductif de concepts, par Antonis Kakas, Fabrizio Riguzzi et Yannis Dimopoulos[DK96, KR00], la relation de couverture se définissant alors *via* une relation de dérivation abductive. Dans ce travail assez riche intégrant induction et abduction, est présenté en particulier le programme *I-ACL*, qui permet de traiter l'incomplétude selon la perspective que nous adoptons de maintien d'un espace des versions correct. L'incertitude sur les faits est prise en charge par des suppositions faites sur des prédicats dits *abductibles*, l'abstraction résultant de la présence, en complément, de prédicats non-abductibles.

Nous avons suivi une voie différente, celle de l'apprentissage par interprétation, dans lequel l'information sur chaque exemple est rangée dans sa propre représentation, et définit l'*apprentissage par possibilités*. Dans un rapport technique, nous avons montré, en adaptant la notion de réduction, que l'*apprentissage par satisfiabilité* se réduisait à l'*apprentissage par possibilités*, et nous avons exploré plusieurs instantiations de celui-ci, ainsi que les conséquences sur les algorithmes implémentant ces formes d'apprentissage[BSV06]. Nous donnons ci-dessous la définition générale de l'apprentissage par possibilités :

Définition 18 (Apprentissage par Possibilités)

Soient HB une base de Herbrand, h une formule, $\{P_i\}$ des théories clausales représentant les possibilités abstraites d'un exemple e . $X \models_{HB} Y$ représente l'inclusion de l'ensemble des modèles de X sur HB dans l'ensemble des modèles de Y sur HB :

- h est compatible⁺ avec l'exemple positif $e = \{P_1, \dots, P_n\}$ ssi
 $\exists P_i \in e$ tel que $P_i \models_{HB} h$
- h est compatible⁻ avec l'exemple négatif $e = \{P_1, \dots, P_n\}$ ssi
 $\exists P_i \in e$ tel que $P_i \wedge h \models_{HB} \square$

Une possibilité abstraite P_i généralise une possibilité e_i , c'est-à-dire une interprétation sur HB décrivant un des exemples complets possibles. Le sens de l'abstraction est que la condition sur h doit être satisfaite quel que soit le modèle de P_i qui est choisi, et revient à considérer que le fait de considérer P_i doit *suffire* à satisfaire la condition.

Exemple Considérons le concept "vole". Chaque exemple est un oiseau *léger* ou *non-léger*, *migrateur* ou *non-migrateur* et soit *rouge* soit *vert*. Une hypothèse est ici une DNF monotone¹, comme par exemple $(\text{léger} \wedge \text{rouge}) \vee (\text{non-léger} \wedge \text{migrateur})$ que nous écrirons plus simplement *léger rouge* \vee *non-léger migrateur*. Soit un exemple positif représentant un oiseau migrateur vert dont le poids est inconnu. Nous supposons de plus qu'en ce qui concerne les oiseaux migrants, l'apprenti sait que la couleur n'intervient pas dans son aptitude au vol, et que donc tous les oiseaux semblables à celui-ci à la couleur près doivent être également reconnus comme positifs². Ceci s'écrit $e = \{P_1, P_2\}$ avec $P_1 = \text{migrateur léger}$, $P_2 = \text{migrateur lourd}$.

L'hypothèse $h = \text{non-migrateur rouge} \vee \text{migrateur}$ convient car $P_1 = \text{migrateur léger} \models_{HB} \text{migrateur}$.

Soit un exemple négatif représentant un oiseau non-migrateur léger dont la couleur est inconnue. Ceci s'écrit $e = \{P'_1, P'_2\}$ avec $P'_1 = \text{non-migrateur léger rouge}$ et $P'_2 = \text{non-migrateur léger vert}$.

0. *Learning from entailment*

1. forme normale disjonctive ne contenant que des littéraux positifs

2. Disons que l'apprenti est un ornithologue particulièrement peu averti.

L'hypothèse $h = (non-migrateur\ rouge) \vee léger$ convient également car $P'_2 = non-migrateur\ vert \wedge (non-migrateur\ rouge \vee migrateur) \models_{HB} \square$.

□

5.1.1 LEa : Apprentissage à partir d'exemples (extrêmement) ambigus

Nous avons récemment travaillé sur LEa^3 [BSV09], un programme d'apprentissage à partir d'exemples ambigus instanciant ces idées dans le cadre suivant :

- Une hypothèse est une forme normale disjonctive monotone (DNF^+) sur un langage propositionnel. La base de Herbrand est ici simplement constituée des atomes du langage.
- On dispose d'une théorie clausale B que doivent satisfaire toutes les instances de l'univers considéré.
- Un exemple est ambigu mais pas abstrait.

Dans LEa , un exemple ambigu est représenté sous forme extensionnelle, par un ensemble d'interprétations satisfaisant B . Un exemple peut initialement être représenté sous forme intensionnelle, par exemple par une théorie clausale $B \wedge \vec{e}$ où \vec{e} est la conjonction des littéraux positifs et négatifs représentant une interprétation partielle : les faits connus sur l'exemple. Nous avons en particulier utilisé cette représentation pour traiter des problèmes de fonctions booléennes dans un univers contraint :

Exemple

Soit une fonction booléenne cible f sur les atomes a_1, \dots, a_9 . Comme LEa apprend des DNF monotones, on ajoute au langage, pour chaque atome a_i , un atome $not-a_i$ de manière à pouvoir apprendre f sans se restreindre aux DNF monotones. La théorie du domaine contient alors toutes les clauses de la forme $(a_i \vee not-a_i)$ ou $\neg(a_i \wedge not-a_i)$.

Nous supposons ici de plus que l'univers des instances possibles n'est pas $\{0, 1\}^9$ mais est contraint : toute instance doit aussi vérifier $B_u = \{a0 \leftarrow a1, a2 \leftarrow a3, a4 \leftarrow a5, a6 \leftarrow a7, a8 \leftarrow a9\}$. La théorie du domaine est donc constituée de B_u et des contraintes de représentations précitées :

$$B = B_u \cup_{i \in 1..9} \{(a_i \vee not-a_i), \neg(a_i \wedge not-a_i)\}.$$

LEa ne considèrera, comme possibilité pour chaque exemple ambigu e , que les modèles de B .

□

LEa construit de manière gloutonne l'ensemble des termes d'une DNF^+ solution, en maximisant à chaque pas, selon une exploration top-down (du général vers le spécifique), la couverture de la DNF^+ courante. Pour chaque terme, l'espace de recherche est limité, comme dans $PROGOL$ [Mug95] aux sous-termes d'un exemple positif (appelé *graine*) encore non couvert \dot{x} ⁴.

LEa apprend une DNF^+ à partir d'exemples ambigus, ce qui le distingue d'autres méthodes gloutonnes sur les points suivants :

1. La compatibilité, positive et négative, remplace la relation de couverture.
2. LEa doit maintenir la cohérence des suppositions faites sur les exemples négatifs aux étapes précédentes. On ne peut se contenter à l'étape i d'évaluer la compatibilité d'un nouveau

3. LEarning from Ambiguous examples

4. Rappelons que \dot{x} est l'unique terme maximale spécifique du langage dont x est un modèle.

terme t_i avec les exemples d'apprentissage, il faut en réalité évaluer la nouvelle hypothèse courante $h_i = t_0 \vee \dots \vee t_{i-1} \vee t_i$, comme le montre l'exemple ci-dessous.

3. La graine est elle-même ambiguë, ce dont il faut tenir compte dans l'exploration des termes.

Exemple Concernant le point 2, considérons l'hypothèse courante $h_1 = a_0a_1 \vee a_1a_2$ compatible avec l'exemple négatif $e = B_u \cup \{a_0; a_2\}$ et le nouveau terme $t = \text{not-}a_1a_2$. t est compatible⁻ avec e en supposant que $\text{not-}a_1$ est faux, cependant $h_2 = a_0a_1 \vee a_1a_2 \vee \text{not-}a_1a_2$ ne l'est pas car il n'y a aucun modèle de e qui ne soit modèle de h_2 . En effet, il faudrait pour cela supposer à la fois $\text{not-}a_1$ pour contredire $a_0a_1 \vee a_1a_2$, et a_1 pour contredire $\text{not-}a_1a_2$, ce qui est impossible car $(a_i \vee \text{not-}a_i)$ est dans la théorie du domaine B_u . \square

Cependant, si le parti pris extensionnel, chaque exemple étant représenté par ses modèles, permet de ramener chaque test de compatibilité à un ensemble de tests d'inclusion, ce n'est acceptable que dans la mesure où le nombre de modèles est faible. Pour réduire ce problème, nous avons utilisé deux remarques :

- En ce qui concerne le test de compatibilité, les modèles minimaux suffisent pour les exemples négatifs, et les modèles maximaux pour les exemples positifs.
- L'ensemble des modèles d'un exemple peut se mettre sous forme d'un produit cartésien.

Ce dernier point amène à utiliser une représentation *multi-table*, au sens défini par Erick Alphonse dans ses travaux sur les macro-opérateurs en programmation logique inductive[Alp04]. Dans le cas particulier des exemples à *valeurs manquantes*, ceci permet un gain exponentiel en le nombre de valeurs manquantes, comme le montre l'exemple ci-dessous.

Exemple

Soit un exemple ambigu e sur un langage défini sur les atomes a_1, \dots, a_n . e est ici simplement un exemple à valeurs manquantes, dans la mesure où la théorie B ne contient que les contraintes de la forme $(a_i \vee \text{not-}a_i)$ et $\neg(a_i \wedge \text{not-}a_i)$: e peut être simplement décrit par les valeurs des atomes connus et un symbole " ? " pour les atomes de valeur inconnue. Nous considérons ici que dans e les 10 premiers atomes ont des valeurs, et que les $n - 10$ autres valeurs sont manquantes, ce qui donne :

$$e = 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, ?, \dots, ?$$

La représentation extensionnelle de e est constituée de 2^{n-10} interprétations. Cependant, cet ensemble s'écrit également sous la forme d'un produit de n tables :

$a_1 \text{ not-}a_1$...	$a_{10} \text{ not-}a_{10}$...	$a_{n-1} \text{ not-}a_{n-1}$	$a_n \text{ not-}a_n$
1 0	...	1 0	...	1 0	1 0
				0 1	0 1

Les 10 premières tables ont une seule ligne, la valeur de l'atome a_i étant connue et déterminant celle de $\text{not-}a_i$. Les $n - 10$ tables suivantes contiennent deux lignes, les deux interprétations partielles possibles sur $\{a_i, \text{not-}a_i\}$ étant $\{a_i\}$ et $\{\text{not-}a_i\}$. Lors du test de compatibilité, on ne considérera donc au plus que n fois 2 interprétations partielles sur des sous-ensembles de 2 atomes. \square

Plus généralement, un exemple ambigu peut toujours se mettre sous la forme d'une théorie clausale⁵. Si on considère que deux atomes apparaissant dans une même clause sont liés, on obtient un graphe dont les sommets sont les atomes, et dont la décomposition en composantes connexes

définit une partition des atomes, qui elle-même définit une partition de la théorie clausale, en théories partielles. Les modèles de la théorie clausale sont alors obtenus comme produit cartésien des modèles des théories partielles. On en déduit la représentation multi-table de l'exemple.

LEa a été expérimenté sur des problèmes booléens pour lesquels la probabilité qu'un atome n'ait pas une valeur déterminé dans un exemple variait de 0 à 0.95. Les expériences ont montré, que *LEa* résiste bien aux probabilités élevées d'incomplétude, et confirmé la convergence théorique de la méthode. La figure 5.1 montre, pour des valeurs de p entre 0 et 0.9, et 3000 exemples ambigus, la justesse obtenue sur un problème booléen contraint (voir l'exemple en début de paragraphe) par *LEa*, ainsi que par l'arbre de décision *C4.5* et le modèle de Bayes naïf. La figure 5.2 montre sur le même problème la justesse en fonction du nombre d'exemples pour une probabilité d'incomplétude $p = 0.9$.

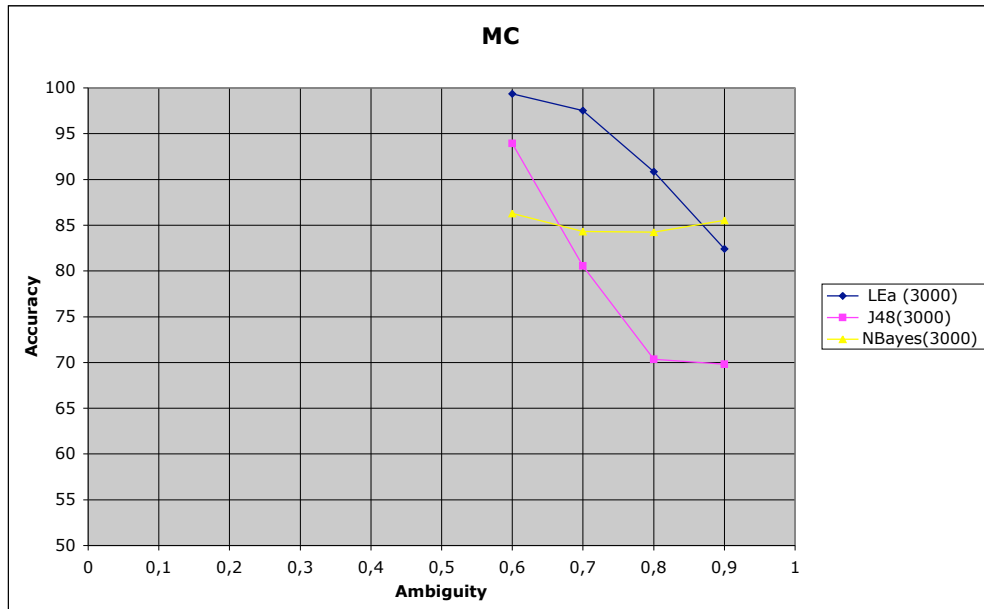


FIGURE 5.1 – Justesse en fonction de la probabilité d'incomplétude en apprentissage, pour le problème contraint *MC* avec 3000 exemples ambigus. Ici la justesse est évaluée sur des exemples test non ambigus. *LEa* résiste mieux que *J48* (implémentation de *C4.5*) à l'incomplétude.

Travaux en cours, et futurs. Nous étendons actuellement *LEa* à l'apprentissage de DNF^+ existentielles en ordre 1, ce qui pose de nouvelles difficultés en ce qui concerne l'efficacité des

5. Si un exemple est sous forme extensionnelle, il est toujours possible de construire une forme normale disjonctive canonique et de la transformer en une forme normale conjonctive approximativement minimale, problème que permettent de traiter les méthodes de simplification de formules booléennes.

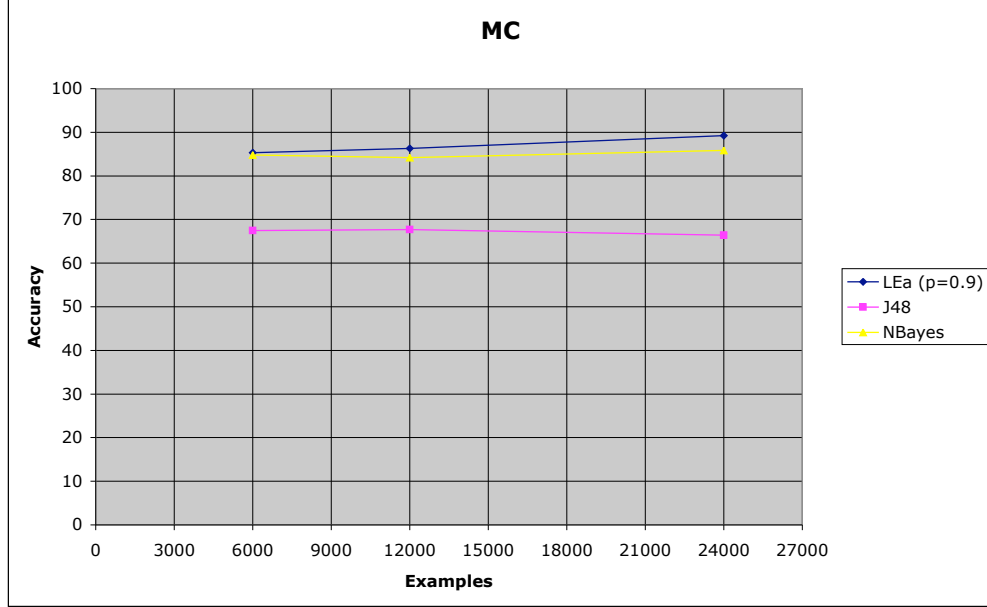


FIGURE 5.2 – Justesse en fonction du nombre d'exemples ambigus, pour le problème contraint MC avec une probabilité d'incomplétude de 0.9. La convergence de *LEa* est lente, mais continue.

tests de compatibilité. Nous nous intéressons également à l'extension des idées présentées ici à une méthode incrémentale en ordre 1 dont la finalité est afférente à l'apprentissage collectif présenté au paragraphe suivant. Un autre travail en cours concerne l'apprenabilité, et plus précisément la complexité d'échantillonnage, c'est à dire le nombre d'exemples ambigus nécessaires pour la convergence PAC au sens de Valiant [Val84], en apprentissage à partir d'exemples ambigus (en ordre 0).

5.2 Apprentissage collectif

Pendant la thèse d'Alejandro Guerra-Hernandez [GH03], nous nous sommes intéressés, avec Amal El Fallah-Segrouchni à l'apprentissage symbolique multi-agents. Le système multi-agent (SMA) considéré est composé d'agents communicants de type *BDI* (Beliefs, Desires, intentions). Pour cela, Alejandro Guerra-Hernandez a défini, et implantée, une architecture BDI incluant un module d'apprentissage basé sur des techniques inductives d'ordre un (précisément l'algorithme Tilde[BR98]). Dans ce travail les agents partagent des plans, chacun associé à un contexte définissant dans quelles conditions le plan peut-être déclenché. Le module d'apprentissage a pour rôle de définir un contexte dans lequel un plan donné devrait réussir, c'est-à-dire se terminer par l'atteinte du but associé au plan. On se ramène donc à un problème d'apprentissage de concept

à partir d'exemples positifs (les succès) et négatifs (les échecs). On considère qu'un agent dispose dans sa mémoire d'un ensemble $E = E^+ \cup E^-$ d'exemples positifs et négatifs de situations de déclenchements du plan. Pour être une solution l'hypothèse h doit être *cohérente* au sens où h doit couvrir les exemples positifs (complétude) et doit ne couvrir aucun exemple négatif (correction). Nous noterons $Cons(h, E)$ la cohérence de h avec E , en tant que cas particulier de *consistance*, notion plus générale que nous utiliserons par la suite.

Exemple Un robot-ouvrier a un plan WALL PAINTING-1 destiné à lui permettre de peindre un mur. Les exemples positifs et négatifs de déclenchement du plan WALL PAINTING-1 sont les suivants⁶ :

- $e_1^+ = \text{wall-is-dry empty-hand enough-paint busy}$
- $e_2^+ = \text{wall-is-dry busy-hand no-paint not-busy}$
- $e_3^- = \text{wall-is-wet empty-hand enough-paint not-busy}$
- $e_4^- = \text{wall-is-dry empty-hand no-paint busy}$

L'hypothèse $h = (\text{wall-is-dry} \wedge \text{enough-paint}) \vee (\text{wall-is-dry} \wedge \text{not-busy})$ est alors cohérente : $Cons(h, E)$ est vrai.

□

Les agents doivent modifier les contextes de déclenchement des plans en fonction des succès et des échecs de leurs actions. Ils sont également pourvus de capacités de communication et peuvent donc envoyer des requêtes aux autres agents. Plusieurs niveaux d'apprentissage sont considérés, selon, par exemple que les agents puissent ou non interpréter les actions d'autres agents pour les inclure dans leur représentation du monde (c'est-à-dire d'eux-mêmes (leur état interne) et de l'environnement) [GHFSS01, GHFSS04a, HSS05, GHFSS04b].

Cependant, si ces agents peuvent ici apprendre en tirant profit de leur capacités de communication, par exemple en s'informant, lorsqu'un contexte de plan devient insatisfaisant, auprès d'autres agents de leurs succès et échecs lors du déclenchement de ce même plan, la société d'agents manque encore d'une approche collective de la révision des connaissances qu'ils partagent. Nous avons alors entamé un travail avec Gauvain Bourgne sur la conception d'un programme expérimentant un protocole d'apprentissage collectif. Ce travail s'est poursuivi, sous forme d'une collaboration pendant sa thèse encadrée à Paris IX par Nicolas Maudet et Suzanne Pinson sur un sujet connexe, et a finalement été intégré à celle-ci. Nous poursuivons actuellement cette voie que je détaille dans les paragraphes suivants.

5.2.1 Révision collective de théorie par maintien de la consistance dans un système multi agent : SMILE;-)

Dans le programme SMILE (Sound Multi-agent Incremental LEarning)[BSS07], un flux d'exemples parvient à la collectivité par l'entremise d'un agent ou un autre et la question est celle du maintien de la consistance (ici la cohérence) de l'hypothèse relativement à l'ensemble des informations (les exemples) dans le système. Une seule hypothèse est maintenue et commune à tous les agents : les

6. On omet ici l'identité du mur, le plan est ici un plan instancié sur le mur à peindre et notre robot-ouvrier, ce qui nous ramène dans cet exemple à une représentation en ordre 0.

6. Sound Multi agent Incremental LEarning

agents sont dits *non-individualistes* et adoptent une hypothèse qui leur est proposée et qu'ils ne peuvent réfuter.

Dans la présentation ci dessous, je généralise le propos en considérant le problème d'apprentissage comme un problème de révision de théorie utilisant une notion générale de consistance, une hypothèse représentant alors une théorie révisable.

Dans SMILE, les agents constituent une société ou système multi-agents en interaction avec un environnement, et rangent chacun des informations, c'est-à-dire des connaissances factuelles non révisables perçues de l'environnement ou reçues d'un autre agent, dans leur mémoire d'information K_i . Chaque agent r_i possède aussi un ensemble de croyances révisables B_i dont une partie B_C est commune avec les croyances des autres agents et forme la *théorie* courante révisable de la société d'agents. Lorsqu'un agent reçoit une *information contradictoire*, il doit réviser B_C de manière à ce qu'elle redevienne consistante avec sa mémoire K_i . De plus, il doit aussi rétablir la *sma-consistance* de B_C , c'est-à-dire la consistance de B_C relativement à l'information distribuée dans la société d'agents. Pour rétablir la *sma-consistance*, un ensemble d'interactions avec les autres agents est nécessaire. L'agent ayant initié la révision prend alors le rôle d'un *apprenti*, les autres agents prenant chacun un rôle de *critique*. Remarquons que les agents peuvent être tour à tour apprentis et critiques, et ne restent pas attachés à un rôle spécifique. En résumé les agents sont autonomes, l'information est distribuée dans la société, et, dans la mesure où des informations peuvent être transmises d'un agent à l'autre, elle est redondante.

Définition 19

- Un agent r_i est a-consistant ssi $Cons(B_i, K_i)$ est vrai. (*consistance individuelle*)
- Un agent r_i est groupe-consistant ssi $Cons(B_i, K)$ est vrai, où $K = \cup_{j \in \{1, \dots, n\}} K_j$ est l'information distribuée dans un sous-ensemble g du n -SMA. En particulier r_i est sma-consistant si g est le n -SMA lui-même⁷ (*consistance globale*).
- Un n -SMA est consistant ssi les agents r_i sont sma-consistants.

Nous supposons que chaque agent peut vérifier sa propriété de consistance individuelle $Cons(B_i, K_i)$ et dispose d'un *mécanisme élémentaire de révision* M . Dans une société non-individualiste, lorsqu'un agent r_i réviser ses croyances B_i en B'_i , changeant alors la théorie commune courante B_C en B'_C , les autres agents r_k doivent également réviser leurs croyances B_k en $B'_k = (B_k - B_C) \cup B'_C$.

Nous supposons dans ce qui suit que la consistance est *additive* et que les croyances communes B_C sont *indépendantes* des croyances particulières de chaque agent, c'est à-dire :

- Si $Cons(B_i, K_1)$ et $Cons(B_i, K_2)$, alors $Cons(B_i, K_1 \cup K_2)$.
- $Cons(B_i, K_i)$ ssi $Cons(B_i - B_C, K_i)$ et $Cons(B_C, K_i)$.

Nous notons M le mécanisme de révision individuel qui permet à un agent r_i recevant une information contradictoire k ⁸ de réviser B_i en $B'_i = M(B_i)$ et de maintenir ainsi sa consistance individuelle. Nous exigeons alors d'un mécanisme de révision M_s qu'il soit *sma-consistant*, au sens où il doit permettre à un agent de maintenir sa sma-consistance. Finalement M_s sera dit

7. un n -SMA est une société de n agents

8. Changeant K_i en $K'_i = K_i \cup k$ tel que $Cons(B_i, K')$ est faux.

fortement sma-consistant si lorsque M_s est appliqué par un agent l'ensemble des agents retrouve sa sma-consistance.

Un mécanisme de révision fortement sma-consistant dans une société d'agents non individualistes

M_s est constitué de révisions individuelles par un agent apprenti et d'interactions avec les agents critiques. Plus précisément :

1. M_s est déclenché par l'agent a lorsqu'il reçoit une information contradictoire k et prend le rôle d'apprenti. Le mécanisme consiste en un ensemble de répétitions du schéma suivant :
 - L'agent a révisé B_C
 - Un ensemble d'interactions $I(a, a_j)$ ont lieu entre l'apprenti et les critiques jusqu'à ce qu'une information contradictoire soit envoyée à a_i (nouvelle itération) ou que tous les critiques ont envoyé un message d'acceptation de la révision (Stop).
2. L'apprenti envoie alors un message de validation de la révision courante B_C .
3. Les agents, étant non-individualistes, *adoptent* alors cette révision, c'est-à-dire remplacent leur version des croyances communes par la révision.

L'interaction $I(a, a_j)$ est constituée des actions suivantes :

- l'agent a envoie la révision B_C à a_j .
- Si $Cons(B_C, E_j)$ est vraie, a_j envoie un message *accept* à a sinon, a_j envoie une information contradictoire e à a ($Cons(B_C, e)$ est faux).

Nous avons montré que ce mécanisme, lorsque les conditions d'additivité et d'indépendance sont vérifiées par la consistance, est fortement sma-consistant : il assure que chaque agent est consistant avec l'ensemble des informations dans la société.

Expériences sur la révision d'hypothèses en apprentissage de concepts

Le mécanisme proposé a été appliqué à l'apprentissage de concept, la théorie se réduisant à une hypothèse pour un concept-cible, représentée sous la forme d'une DNF^+ propositionnelle, comme au paragraphe 5.1.1, les mémoires d'informations étant des mémoires d'exemples E_i , et la consistance à maintenir étant ici la cohérence de H avec l'ensemble des exemples envoyés par l'environnement aléatoirement aux agents. Remarquons que, dans les expériences présentées ici, l'apprentissage est *lent* au sens où on suppose que lorsqu'un exemple est envoyé par l'environnement, une révision complète a le temps d'être effectuée dans la société d'agents avant qu'un autre exemple ne soit envoyé à un autre agent. Nous appellerons dans ce cadre *contre-exemples*, les informations contradictoires, c'est à dire les exemples rendant incohérente l'hypothèse courante.

Le mécanisme de révision individuel M , qui étant donné un contre-exemple e parvenant à un agent lui permet de produire une hypothèse révisée H' cohérente avec le nouvel état de la mémoire $E_i \cup e$ est ici une méthode incrémentale ascendante et dirigée par les données développée par Mhammet Henniche (voir 2.2.2). Plus précisément il s'agit dans les expériences de la variante *iMind2*.

iMind2 exécute une révision minimale de H à partir d'un ensemble d'exemples E (la mémoire) et d'un ou plusieurs exemples contradictoires e_1, \dots, e_n . Dans le cas d'un seul exemple contradictoire e la révision suit le schéma suivant :

- Si H ne couvre pas $e = e^+$ (e^+ est un *contre-exemple positif*), H est révisée en généralisant de manière minimale un des termes h_i composant H , c'est à dire $h'_i = \text{lgg}(h_i, e)$ ou, si aucun terme correct n'est ainsi obtenu, en ajoutant un nouveau terme $h = \text{lgg}(e)$ à H .
- Si H couvre $e = e^-$ (e^- est un *contre-exemple négatif*), chaque terme h couvrant e^- est éliminé de H et remplacé par un ensemble de nouveaux termes $\{h'_1, \dots, h'_n\}$ corrects relativement à $E^- \cup \{e\}$. De manière à assurer la complétude, la disjonction de ces termes doit couvrir les exemples de E^+ qui ne sont pas couverts par $H - \{h\}$ ⁹.
- H est rendue *non-redondante* au sens suivant : tout terme h plus spécifique qu'un autre terme h' est éliminé.

Ce mécanisme est donc *semi-incrémental* puisqu'il utilise les exemples de la mémoire de l'agent, et fournit une hypothèse minimalement révisée et non redondante. En particulier ce dernier point est ce qui permet d'une révision à une autre de parfois diminuer la taille (en nombre de termes) de l'hypothèse courante.

SMILE a été expérimenté en mode *séquentiel minimal* dans lequel l'agent appreni envoie une révision à un seul critique (séquentiel) qui renvoie un seul contre-exemple même s'il a plusieurs exemples contradictoires (minimal)¹⁰. En figure 5.3 on trouvera un exemple simple d'exécution du mécanisme de révision collective dans une société de trois agents lorsque l'un des agents doit réviser l'hypothèse commune.

Nous avons mené des expériences sur plusieurs fonctions booléennes plus ou moins difficiles à apprendre ainsi que sur plusieurs problèmes types de l'UCI Database [DNM98]. Parmi les fonctions booléennes, le *multiplexer-11 M11* une fonction de 8 termes de 3 littéraux sur 11 atomes, et une fonction *xor* (*Xor3_25*) qui prend la valeur *vrai* si le nombre d'atomes dont la valeur est 1 parmi 3 d'entre eux, est impair (il y a en tout 25+3 atomes).

Principaux résultats En ce qui concerne les ressources consommées pendant l'apprentissage, on trouve un temps d'exécution qui est grossièrement linéaire en fonction du nombre d'agents. De même la *redondance* d'information, mesurée par le rapport du nombre total d'exemples dans les mémoires d'agents au nombre d'exemples envoyés par l'environnement, est linéaire : il y a un pic de redondance dans la partie la plus active de l'apprentissage, puis celle-ci décroît lentement vers sa valeur minimale 1 : lorsqu'il n'y a plus de contradictions les seuls exemples mémorisés par un agent sont ceux reçus de l'environnement.

Les résultats les plus intéressants observés sur les fonctions booléennes concernent la qualité de l'hypothèse apprise. Celle-ci est mesurée par sa *taille* (en nombre de termes) représentant sa complexité, ainsi que par la *justesse*, estimant sur un ensemble d'*exemples test* la probabilité de prédiction correcte du label (+ ou -) d'une instance inconnue :

- Une société de n agents apprend une hypothèse plus juste qu'un agent isolé ayant reçu les mêmes exemples.

Ceci est d'autant plus vrai que le problème est difficile. La figure 5.4 montre l'évolution de la

9. La variante *iMind2* utilise un algorithme glouton qui considère l'ensemble E_h des exemples de E^+ couverts uniquement par h , spécialise de manière minimale h en h' de manière à le rendre correct, ajoute à H $h_1 = \text{lgg}(E_{h_1})$, cherche une nouvelle spécialisation h' de h en considérant les éléments encore non couverts, et ainsi de suite jusqu'à ce que tous les éléments de E_h aient été couverts.

10. Dans un travail en cours nous expérimentons *SMILE* en mode *broadcast maximal* : l'appreni envoie sa révision H à tous les critiques et ceux-ci renvoient tous leurs contre-exemples de H .

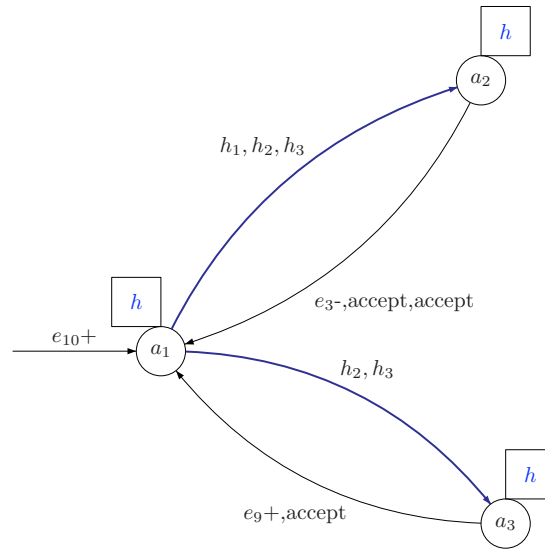


FIGURE 5.3 – Un exemple de révision dans SMILE dans sa version séquentielle. Les trois agents partagent la même hypothèse h . Un contre-exemple positif e_{10} est alors perçu par l'agent a_1 qui devient apprenti. La suite des interactions est la suivante : - a_1 révisé h en h_1 à a_2 qui lui renvoie e_3^- comme contre-exemple négatif de h_1 (h_1 couvre e_3^-) - a_1 propose alors h_2 qui est accepté par a_2 - a_1 propose alors h_2 à a_3 qui renvoie le contre-exemple positif e_9^+ à a_1 (h_2 ne couvre pas e_9^+) - a_1 propose alors la révision h_3 à a_3 qui l'accepte -cette révision est alors proposée à a_2 qui l'accepte également. L'hypothèse est maintenant consistante avec l'ensemble des exemples dans la société de trois agents. a_1 envoie alors un message de validation à a_2 et a_3 . A la réception de ce message a_2 et a_3 adoptent h_3 comme hypothèse courante. h_3 est alors la nouvelle hypothèse courante commune aux trois agents.

justesse pour le problème *xor_3_25*, en fonction du nombre d'exemples reçu dans le système pour des sociétés de n agents, n variant de 1 à 20.

- Une société de n agents apprend une hypothèse plus simple qu'un agent isolé recevant les mêmes exemples de l'environnement. La figure 5.5 montre, dans le même problème, l'évolution de la taille de l'hypothèse.

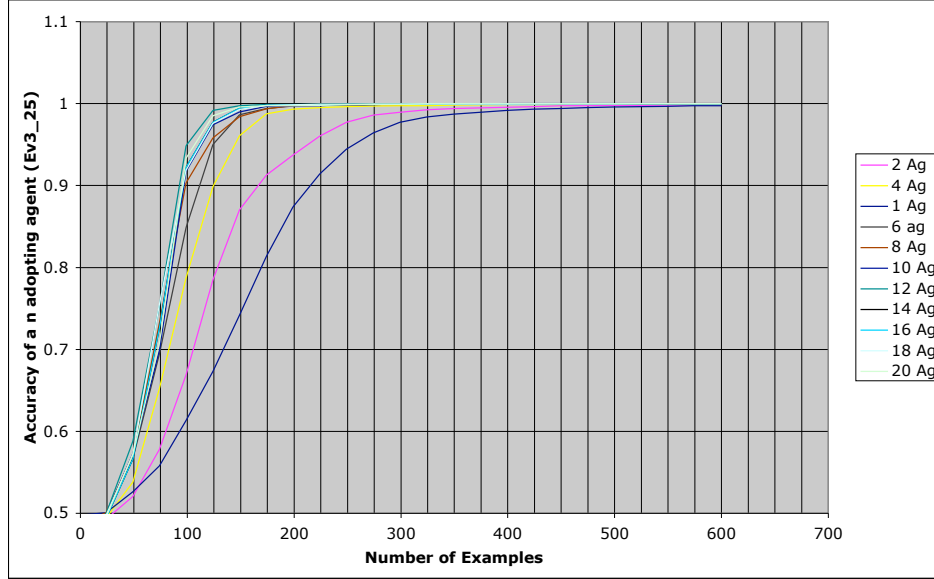


FIGURE 5.4 – Justesse obtenue sur le problème *xor_3_25* par des sociétés de 1 à 20 agents non-individualistes. Sur ce problème difficile la justesse lorsque la société a reçu 100 exemples de l'environnement passe de ≈ 0.61 (un agent) à ≈ 0.95 (20 agents).

Cette efficacité dans l'exploitation des exemples d'apprentissage, impressionnante pour certaines fonctions difficiles comme le *xor* est due principalement à deux effets :

- un effet de *sélection* par les agents critiques des exemples dans la mémoire de chaque agent : plus le nombre d'agents est grand et plus faible est la proportion d'exemples reçus par un agent directement de l'environnement et donc n'ayant pas toujours déclenché de révision.
- un effet d'*exploration intensive* de l'espace d'hypothèses : Lorsque n est grand chaque agent révisé à partir de peu d'exemples, et il y a un grand nombre d'interactions, et donc de révisions. De plus chaque agent part d'une hypothèse adoptée, et la révisera en fonction de ses propres exemples, ce qui accroît la diversité de l'exploration.

Ce dernier effet semble d'après de récentes expériences plus important (il contribuerait à hauteur de 2/3 de l'amélioration totale).

D'une certaine manière cette amélioration de l'hypothèse trouvée représente une forme d'émergence d'une exploration approfondie de l'espace de recherche, associée aux nombreuses interactions dans la société d'agents : en effet, rien dans le comportement des agents n'est dirigé directement vers l'optimisation collective de l'hypothèse : celle-ci est un effet global résultant de nombreux ajustements locaux.

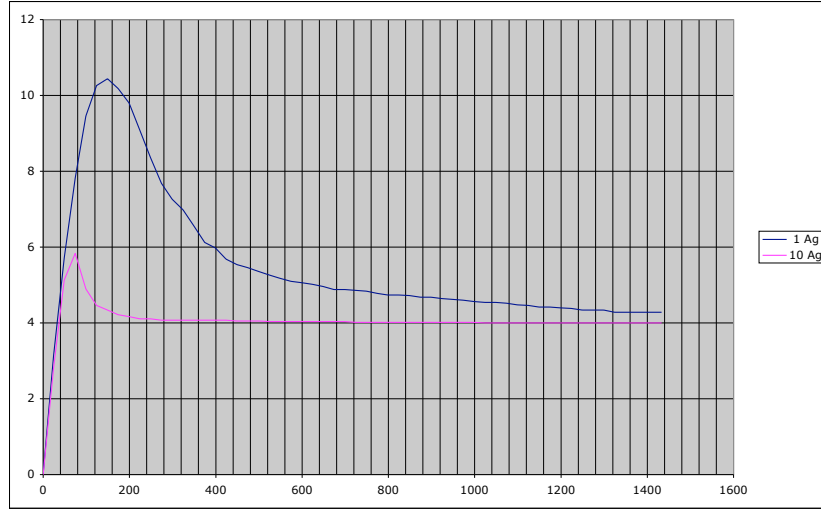


FIGURE 5.5 – Taille de l’hypothèse construite pour le problème *xor-3-25* un agent isolé et une société de 10 agents. Ici la société de 10 agents converge plus vite vers la représentation la plus simple (4 termes) de la solution. Le pic de complexité se situe vers 100 exemples, c’est à dire là où l’apprentissage est le plus actif, et où la différence en justesse est la plus importante entre une société d’agents et un agent isolé.

Il faut cependant remarquer que les expériences sur les problèmes classiques de l’UCI, cette amélioration n’est le plus souvent pas perceptible : pour ces problèmes la difficulté n’est pas de trouver une hypothèse optimalement simple et cohérente dans un espace dans lequel il est difficile de se diriger, et de ce fait, une exploration intensive perd son utilité.

5.2.2 Révision collective de théorie dans une société d’agents organisée en un réseau

Le mécanisme de révision de SMILE a été étendu récemment au cas où les agents ne communiquent pas tous directement, mais sont organisés en un graphe (en général connexe) : chaque agent a un groupe de voisins avec lesquels il peut communiquer. L’application du mécanisme M_s de SMILE ne garantit plus la consistance de l’agent relativement à la société mais seulement relativement au groupe de ses voisins. Pour retrouver la consistance de la société, il y a deux approches principales :

- Soit l’apprentissage reste *centré* sur l’agent apprenant ayant reçu de l’environnement le contre-exemple causant une révision. Dans ce cas les critiques *relaient* l’hypothèse à leurs voisins et les contre-exemples éventuels à l’agent apprenant.
- Soit l’apprentissage est *distribué* dans la société : lorsqu’un agent, propageant l’hypothèse de l’apprenant, reçoit un contre-exemple d’un de ses voisins, il devient le nouvel apprenant (l’apprentissage est distribué dans la société via une *racine variable* de l’apprentissage.).)

Ces deux approches ont été expérimentées sur différentes topologies de réseau [BFSMS08, BESS09]. Les deux approches maintiennent les résultats de SMILE concernant la justesse et la simplicité des hypothèses acquises. L'approche centrée est efficace, en ce qui concerne en particulier la redondance, mais tend à accroître les coûts de communication, et peut devenir impraticable lorsque le réseau devient grand. L'approche distribuée est plus économe. Surtout cette approche distribuée est un bon candidat pour une propagation non séquentielle des hypothèses : en effet, si dans la société pleinement communicante abordée par SMILE la différence entre propagation séquentielle et multiple (broadcast) ne semble pas primordiale, la propagation multiple est à la fois plus réaliste et pose des problèmes de conflits et de priorités qui restent à aborder lorsque la société est structurée en réseau. De plus, l'approche distribuée a l'avantage de pouvoir s'exprimer comme une propagation due à des comportements purement locaux : il suffit de considérer qu'un agent doit toujours s'assurer que son hypothèse courante est consistante avec ses voisins. Dans ce cas un agent recevant, dans son rôle de critique, d'un autre agent, dans un rôle d'apprenti, un message de validation d'une hypothèse (l'hypothèse h est alors consistante avec tous les voisins de l'apprenti) doit s'assurer avant d'adopter cette hypothèse de sa consistance avec ses propres voisins. Mis à part les aspects protocolaires (choix entre séquentialité et propagation multiple) il n'y a rien à ajouter pour assurer la propagation de l'hypothèse et le maintien de la consistance dans la société d'agents. Autrement dit, on revient à un comportement local de l'agent : ici déclenché par la double nécessité de maintenir sa consistance avec ses voisins et d'adopter une hypothèse validée par un voisin.

5.2.3 iSMILE : révision collective dans une société d'agents individualistes

Nous avons également travaillé sur la variante individualiste de SMILE, dans le cas d'une société pleinement communicante. Le mécanisme de révision collective ne change qu'en un point : lorsqu'un agent critique reçoit un message de validation d'une hypothèse, il n'adopte pas celle-ci. Les conséquences théoriques et expérimentales de cette modification sont cependant importantes [BBESS09]. Une conséquence immédiate est que chaque agent aura sa propre hypothèse courante. On constate expérimentalement, une décroissance sévère de la justesse avec le nombre d'agents. Elle provient d'un effet de *désynchronisation* : l'application du mécanisme de révision collective, privé de l'adoption, ne garantit que la consistance de l'hypothèse de chaque agent a_i avec l'état de la mémoire de la société à l'instant t_i de sa dernière révision. Cependant d'une part on peut limiter cet effet en ajoutant un comportement de resynchronisation aux agents, et d'autre part on peut tirer parti de la diversité des hypothèses dans la société pour utiliser une forme de décision collective, ce qui correspond pour la société d'agent à une forme d'*apprentissage par ensemble* [Die00]. Les expériences menées conduisent aux résultats suivants :

- Redondance et temps d'exécution sont plus élevés que dans le cas non-individualiste, mais restent approximativement linéaires avec le nombre d'agents.
- La resynchronisation ajoute un surcoût élevé en début d'apprentissage mais lorsqu'on s'approche de la convergence le coût global est faible (les hypothèses étant meilleures assez tôt, relativement au nombre d'exemple, elles nécessitent moins de révisions par la suite) et la redondance chute aussi brutalement. De plus l'amélioration en justesse est très nette.

- La décision collective par vote majoritaire améliore également les justesses observées. Le meilleur résultat est obtenu en combinant resynchronisation et décision collective.
- Sur certains problèmes de L'UCI, la décision collective améliore la justesse.

En résumé on perd en partie l'amélioration de la justesse obtenue dans le cas non-individualiste. On garde le bénéfice de l'effet de sélection des exemples, mais on perd celui de l'exploration intensive de l'espace d'hypothèses. On perd également la propriété de consistance globale à la fin de la révision, ce que l'on peut atténuer avec un comportement de resynchronisation qui se révèle peu coûteux. En revanche on bénéficie de la diversité des hypothèses dans la société, ce qui pour certains types de problèmes est précieux.

5.2.4 Adaptation collective dans un réseau de télécommunications

J'ai assisté Francine Krief dans l'encadrement du travail de thèse, bientôt soutenu, de Maïssa M'Baye sur l'autoconfiguration collective des équipements d'un réseau de communication en vue d'optimiser la Qualité de service. Le travail est inspiré des protocoles et comportements proposés dans les travaux ci-dessus, mais avec des points originaux découlant de la nature du problème posé. Par exemple, on est ici directement dans une situation d'adaptation des actions à mener en fonction de l'état perçu de l'environnement, ce qui a été abordé par Maïssa en faisant par exemple varier les labels des exemples (des couples état-action) en fonctions des observations perçues, ou transmises par d'autres entités (ici des équipements) et en limitant la transmission des informations à un voisinage de manière à tenir compte des spécificités des aspects géographiques : quelle est la portée que doit avoir une observation ? L'implémentation d'un simulateur a permis d'expérimenter la proposition, qui s'est révélée étonnement efficace[MKS09]. Ce travail, sur une thématique proche, se démarque des travaux sur l'apprentissage collectif par renforcement [BBDS08][PL05] en prenant le parti de la construction de règles explicites liant états et actions, et se situe dans l'idée de donner des comportements "intelligents" autonomes aux équipements dans un réseau.

5.2.5 Perspectives.

Nous continuons actuellement à travailler sur ces modèles d'apprentissage collectif. Parmi les points abordés :

- La question de la révision non-individualiste distribuée dans un réseau en abandonnant la séquentialité nécessite une étude approfondie. En particulier, plusieurs hypothèses concurrentes circulant dans le réseau, la rationalité des agents dans leur choix des hypothèses à adopter ou à propager devra s'appuyer sur des informations plus fines : on peut ainsi envisager de transmettre en même temps qu'une hypothèse, un historique de celle-ci, contenant l'identité des agents l'ayant révisée ou propagée.
- On ne peut éviter la question de la fusion des croyances, si on veut échapper à l'opposition entre le modèle individualiste et le modèle non individualiste : non seulement un raisonnement peut permettre un choix sur l'attitude, mais l'agent peut aussi chercher à utiliser une hypothèse propagée pour réviser la sienne. La question des formes que peut prendre une telle *réflexion* (pour ainsi nommer une forme de révision non imposée par une contradiction) est ouverte.

- La question de la propagation des informations (exemples), comme conséquence directe, et en sens inverse, de la propagation d’hypothèses, mérite qu’on s’y attarde. En particulier il sera intéressant de voir l’effet d’une distribution hétérogène des exemples dans un réseau. D’ailleurs dans ce cas la question de la diversité des hypothèses prend un autre sens : est-il utile que chaque agent ait la même théorie, celle-ci doit-elle être dépendante des informations que l’agent est susceptible de rencontrer ?
- La question de la structure de la société, abordée ici par une notion de voisinage, peut également être abordée par des structururations granulaires : si une société est structurée en ”tribus“, deux agents d’une même tribu sont-ils plutôt non individualistes ? comment les théories se propagent-elles d’une tribu à l’autre ?
- Du point de vue théorique, l’apprentissage collaboratif par maintien de la consistance proposé ici, a des liens étroits avec l’apprentissage à partir de requêtes dont le cadre a été défini par Dana Angluin[Ang04] et plus précisément avec la révision de théories à partir de requêtes [GSST04]. Les agents critiques définis ici jouent le rôle d’*oracles incomplets* répondant à des *requêtes d’équivalences*. Peut-on envisager une étude théorique de l’apprentissage par requêtes dans un cadre collectif ? ou plus simplement peut-on recentrer les résultats obtenus dans ce contexte pour en tirer parti dans notre approche pragmatique de l’apprentissage collectif ? Une extension naturelle d’une telle modélisation est de considérer l’apprentissage actif, c’est-à-dire l’investigation expérimentale par un agent définissant des expériences destinées à obtenir les réponses à ses *requêtes d’appartenance*¹¹.
- Ce dernier point a déjà été soulevé par Jean Sallantin et ses collègues[ADL⁺07], dans un travail visant à modéliser le processus de découverte scientifique dans sa dimension collective. Cette dernière formalisation est par ailleurs plus ambitieuse que ce que nous avons proposé et expérimenté ici, va dans le sens des travaux de thèse de Gauvain Bourgne[Bou08] et a le mérite de proposer une formulation en logique modale des interactions entre agents.

11. Dans une requête d’appartenance, l’apprenti soumet à l’oracle une instance choisie par lui en vue d’obtenir son label.

Chapitre 6

Conclusion ?

Les perspectives d'un travail de recherche sont par nature aventureuses. Comme dans un jeu vidéo, caractères blancs sur fond vert, le paysage ne sort de l'ombre de la Moria que lorsqu'on avance dans l'obscurité, Galgals et Jaberwockys menacent, mais jouerait-on si tout était prévisible ?

La réflexion sur l'apprentissage en situation d'ambiguïté me paraît un préalable nécessaire pour approcher l'apprentissage collectif non omniscient : c'est l'état naturel d'une entité que de n'avoir qu'une perception partielle d'une observation, la question de l'apprentissage collectif ne se conçoit donc qu'avec des descriptions partielles distribuées dans la société. Si on ignore le fait que des informations portent sur la même observation (disons le même exemple), le modèle proposé de révision collective convient. Cependant on pourrait tirer parti de ces liens mais cela suppose des formes d'interactions plus complexes que celles dont il a été question dans le dernier chapitre. Ainsi si un objet, exemple négatif d'un certain concept, est perçu comme *bleu ou rouge* par a_1 et comme *vert ou bleu* par a_2 , l'objet est *bleu* pour un observateur omniscient sans que ni a_1 ni a_2 le sache. L'hypothèse $h = \text{bleu}$, que a_1 peut formuler, ne peut être réfutée par a_2 , sauf à affiner l'interaction. Une perspective directe des travaux évoqués au chapitre précédent conduit donc, à la fois, à concevoir et expérimenter une méthode de révision en ordre 1 inspirée de *LEa* et à pousser plus loin les modalités d'interactions inductives et abductives entre entités dans une société d'agents. En tout état de cause, il me semble intéressant d'insérer des mécanismes d'induction, d'abduction et de raisonnement dans une entité, située dans un environnement dont elle se distingue, et d'intégrer l'autonomie qui en résulte, y compris vis à vis d'autres entités, comme une caractéristique qui la définit. Dans ce contexte, apprendre et raisonner à différents niveaux d'abstraction me semble également inévitable. Cependant, sur ce point, il me semble que la réflexion n'est pas encore suffisamment avancée. La proposition faite ici d'une abstraction ordonnée intensionnellement et extensionnellement, me semble pouvoir être poussée dans deux directions : d'une part, elle élargit la question de la sélection du langage d'assertions \mathcal{L} à explorer, en la question du choix de l'abstraction utile d'un langage détaillé, question que l'on retrouve dans la réflexion sur les macro-opérateurs menée par Erick Alphonse, ainsi que dans les travaux en cours avec Céline Rouveirol et Sébastien Guérif sur des formes non classiques d'abstraction intensionnelle, d'autre part, elle pose la question de la granularité de l'objet, et de son impact sur l'induction et la déduction. Des travaux en cours avec Véronique Ventos visent à établir les propriétés logiques des extensions Alpha, et plus généralement des extensions abstraites. Les

extensions Alpha faisant apparaître une forme particulière d'exception en négatif (les mammifères en général ne volent pas, on peut donc tenir la chauve-souris comme exceptionnelle en ce qu'elle possède une propriété que par défaut elle ne devrait pas avoir) la question se pose aussi des liens avec des travaux précédents menés avec Pierre Brézellec et Véronique Ventos sur l'effet sur l'apprentissage de l'ajout de modalités *défaut* et *exception* au langage [VBS01].

Les travaux présentés dans ce mémoire sur l'aide à la découverte en Biologie, par le biais de méthodes de recherche de motifs répétés, posent différentes questions : comment ces techniques, extensionnelles et par niveaux, s'articulent-elles avec les méthodes de fouilles de données basées sur la maximalité ? peut-on les étendre à des structures arborescentes, ou de graphes, graphes de systèmes complexes ou graphes sociaux comme ceux explorés par Céline Rouveirol et Rushed Kanawati, en s'inspirant des travaux récents sur la maximalité dans ce domaine ? En voyant plus loin, les premiers travaux présentés ici sur l'apprentissage collectif permettront-ils d'aborder la question de la fouille décentralisée des données ?

Mais le plus intéressant est ce qui est encore dans l'ombre, et donc peut-être pressenti, mais pas décrit, pas plus que les nouveaux problèmes qui en émergeront. Pourquoi jouer sinon ?

Bibliographie

- [ADL⁺07] Mohammad Afshar, Christopher Dartnell, Dominique Luzeaux, Jean Sallantin, and Yannick Tognetti. Aristotle's square revisited to frame discovery science. *JCP*, 2(5) :54–66, 2007.
- [All83] James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11) :832–843, 1983.
- [Alp04] Érick Alphonse. Macro-operators revisited in inductive logic programming. In Rui Camacho, Ross D. King, and Ashwin Srinivasan, editors, *ILP*, volume 3194 of *Lecture Notes in Computer Science*, pages 8–25. Springer, 2004.
- [Ang04] Dana Angluin. Queries revisited. *Theor. Comput. Sci.*, 313(2) :175–194, 2004.
- [AU07] Hiroki Arimura and Takeaki Uno. An efficient polynomial space and polynomial delay algorithm for enumeration of maximal motifs in a sequence. *Journal of Combinatorial Optimization*, 13(3) :243–262, 04 2007.
- [BBDS08] L. Buşoniu, R. Babuška, and B. De Schutter. A comprehensive survey of multi-agent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C : Applications and Reviews*, 38(2) :156–172, March 2008.
- [BBESS09] Gauvain Bourgne, Dominique Bouthinon, Amal ElFallah-Seghrouchni, and Henry Soldano. Collaborative concept learning : Individualistic vs non-individualistic agents. In *21st International Conference on Tools with Artificial Intelligence (IC-TAI'09)*, volume to appear, Newark, NJ, USA, 2009. IEEE Computer Society Press.
- [BBOV09] Radim Belohlávek, Bernard De Baets, Jan Outrata, and Vilém Vychodil. Inducing decision trees via concept lattices. *International Journal of General Systems*, 38(4) :455–467, 2009.
- [BBW98] Z. Bonikowski, E. Bryniarski, and U. Wybraniec-Skardowska. Extensions and intentions in the rough set theory. *Information Sciences : An International Journal*, 107 :149–167, 1998.
- [BD00] H.H. Bock and E. Diday. *Analysis of Symbolic Data*. H.H. Bock and E. Diday, Springer Verlag, 2000.

- [BESS09] Gauvain Bourgne, Amal ElFallah-Seghrouchni, and Henry Soldano. Learning in a fixed or evolving network of agents. In *IEEE / WIC / ACM International Conference on Intelligent Agent Technology*, volume to appear, Milano, Italy, 2009. IEEE Computer Society.
- [BFSMS08] Gauvain Bourgne, Amal El Fallah-Seghrouchni, Nicolas Maudet, and Henry Soldano. Multiagent incremental learning in networks. In The Duy Bui, Tuong Vinh Ho, and Quang-Thuy Ha, editors, *PRIMA*, volume 5357 of *Lecture Notes in Computer Science*, pages 109–120. Springer, 2008.
- [BFV07] Guillaume Blin, Guillaume Fertin, and Stéphane Vialette. Extracting constrained 2-interval subsets in 2-interval sets. *Theoretical Computer Science*, 385(1-3) :241–263, 2007.
- [Bir73] Garrett Birkhoff. *Lattice Theory*. American Mathematical Society Colloquium Publications, Rhode Island, 1973.
- [BM70] M. Barbu and M. Monjardet. *Ordre et Classification, Algèbre et Combinatoire 2*. Hachette Université, 1970.
- [Bou96] Dominique Bouthinon. *Apprentissage à partir d'exemples ambigus*. Informatique, Université Paris 13, December 1996.
- [Bou08] Gauvain Bourgne. *Propagation et raffinement d'hypothèse entre agents délibératifs sous contraintes communicationnelle*. Informatique, université Paris 9, March 2008.
- [BR98] Hendrik Blockeel and Luc De Raedt. Top-down induction of first-order logical decision trees. *Artif. Intell.*, 101(1-2) :285–297, 1998.
- [BS93a] Pierre Brézellec and Henry Soldano. Éléna : A bottom-up learning method. In *ICML*, pages 9–16. Morgan Kaufmann, 1993.
- [BS93b] Pierre Brézellec and Henry Soldano. SAMIA : a bottom-up learning method using a simulated annealing algorithm. In *Proceedings of the European conference on Machine Learning*, volume 667 of *Lecture notes in Artificial Intelligence*, pages 297–309. Springer-verlag, 1993.
- [BS94] P. Brezellec and Henry Soldano. Improvement of the exploration of the search space of a top-down algorithm : Theoretical and experimental results. In *Proc. of the 11th ECAI*, pages 458–462, Amsterdam, The Netherlands, 1994. Wiley and Sons.
- [BS98a] Dominique Bouthinon and Henry Soldano. An inductive logic programming framework to learn a concept from ambiguous examples. In *European Conference on Machine Learning (ECML)*, number 1398 in *Lecture Notes in Artificial Intelligence*, pages 238–249, Chenmitz, 1998.
- [BS98b] P. Brézellec and Henry Soldano. Tabata : a learning algorithm performing a bidirectional search in a reduced search space using a tabu stratégie. In *Eur. Conf. in Art. Int., ECAI-98*, pages 420–424, Brighton, England, 1998. Wiley and Sons.

- [BS99] Dominique Bouthinon and Henry Soldano. A new method to predict the consensus secondary structure of a set of unaligned RNA sequences. *Bioinformatics*, 15(10) :785–798, 1999.
- [BSS07] Gauvain Bourgne, Amal El Fallah Seghrouchni, and Henry Soldano. Smile : Sound multi-agent incremental learning;-). In *International Conference on Autonomous Agents and Multiagent Systems, (AAMAS)*, pages 164–171, Honolulu, Hawaï, 5 2007.
- [BSV99] Dominique Bouthinon, Henry Soldano, and Véronique Ventos. Aspects théoriques et pratiques de l'apprentissage en cas d'incomplétude des données : étendre l'apprentissage par satisfiabilité. In *Actes de la Conférence d'apprentissage CAP99-AFIA*, pages 145–152, Ecole Polytechnique, Palaiseau, 1999.
- [BSV06] Dominique Bouthinon, Henry Soldano, and Veronique Ventos. Logical settings for concept learning from incomplete examples in first order logic. Technical Report CoRR abs/cs/0607096 : (2006), L.I.P.N, 2006.
- [BSV09] Dominique Bouthinon, Henry Soldano, and Veronique Ventos. Concept learning from (very) ambiguous examples. In Petra Perner and Atsushi Imiya, editors, *MLDM*, volume to appear of *Lecture Notes in Computer Science*. Springer, 2009.
- [Cat97] Gianpiero Cattaneo. Generalized rough sets (preclusivity fuzzy-intuitionistic (bz) lattices). *Studia Logica*, 58(1) :47–77, 1997.
- [CB91] P. Clark and R. Boswell. Rule induction with CN2 : Some recent improvements. In *Proc. Fifth European Working Session on Learning*, pages 151–163, Berlin, 1991. Springer.
- [CBS95] Marc Champesme, Pierre Brézellec, and Henry Soldano. Empirically conservative search space reductions. In L. De Raedt, editor, *ILP95*, pages 387–402. DEPTCW, 1995.
- [CC05] Gianpiero Cattaneo and Davide Ciucci. Algebraic structures for rough sets. In *Transactions on Rough Sets II*, volume 3135 of *Lecture Notes in Computer Science*, pages 208–252. springer, 2005.
- [CC08] Gianpiero Cattaneo and Davide Ciucci. A hierarchical lattice closure approach to abstract rough approximation spaces. In *Rough Sets and Knowledge Technology*, volume 5009 of *LNCS*, pages 363–370. Springer, 2008.
- [CFRD07] P. Cellier, S. Ferré, O. Ridoux, and M. Ducassé. A parameterized algorithm for exploring concept lattices. In S.O. Kuznetsov and S. Schmidt, editors, *Int. Conf. Formal Concept Analysis*, LNCS 4390. Springer, 2007.
- [CFRD08] P. Cellier, S. Ferré, O. Ridoux, and M. Ducassé. A parameterized algorithm to explore formal contexts with a taxonomy. *Int. J. Foundations of Computer Science (IJFCS)*, 19(2) :319–343, 2008.

- [CH94] William W. Cohen and Haym Hirsh. Learning the classic description logic : Theoretical and experimental results. In *In Principles of Knowledge Representation and Reasoning : Proceedings of the Fourth International Conference (KR94)*, pages 121–133. Morgan Kaufmann, 1994.
- [Cha95] Marc Champesme. Using empirical subsumption to reduce the search space in learning. In *Conceptual Structures : Applications, Implementation and Theory*, volume 954 of *LNCS*, pages 188–201. Springer, 1995.
- [CHL⁺08] Maxime Crochemore, Danny Hermelin, Gad Landau, Dror Rawitz, and Stéphane Vialette. Approximating the 2-Interval Pattern problem. *Theoretical Computer Science*, 395(2-3) :283–297, 2008. Special issue in honor of the 60th Birthday of Professor Alberto Apostolico.
- [CLM07] Nathalie Caspard, Bruno Leclerc, and Bernard Monjardet. *Ensembles ordonnés finis : concepts, résultats, usages*. Mathématiques et Applications. SPRINGER, 09 2007.
- [CM03] Nathalie Caspard and Bernard Monjardet. The lattices of closure systems, closure operators, and implicational systems on a finite set : a survey. *Discrete Appl. Math.*, 127(2) :241–269, 2003.
- [Coh95] William W. Cohen. Fast effective rule induction. In Armand Frieditis and Stuart Russell, editors, *Proc. of the 12th International Conference on Machine Learning*, pages 115–123, Tahoe City, CA, July9–12, 1995. Morgan Kaufmann.
- [Dan00] Antoine Danchin. A brief history of genome research and bioinformatics in France . *Bioinformatics*, 16(1) :65–75, 2000.
- [DD99] Christophe Dousson and Thang Vu Duong. Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. In Thomas Dean, editor, *IJCAI*, pages 620–626. Morgan Kaufmann, 1999.
- [De 97] Luc De Raedt. Logical settings for concept-learning. *Artif. Intell.*, 95(1) :187–201, 1997.
- [Did01] Edwin Diday. Analyse des données symboliques : un cadre théorique et des outils. In *Tutoriels EGC'2001*, pages 83–149, Nantes, 17-19 janvier 2001.
- [Die00] Thomas G. Dietterich. Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857 :1–15, 2000.
- [DK96] Y. Dimopoulos and A. Kakas. Abduction and inductive learning. In L. De Raedt, editor, *Advances in Inductive Logic Programming*, pages 144–171. IOS Press, 1996.
- [DLLP97] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.*, 89(1-2) :31–71, 1997.

- [DNM98] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.
- [Dom96] Pedro Domingos. Unifying instance-based and rule-based induction. *Machine Learning*, 24(2) :141–168, 1996.
- [Dou02] Christophe Dousson. Extending and unifying chronicle representation with event counters. In Frank van Harmelen, editor, *ECAI*, pages 257–261. IOS Press, 2002.
- [DP02] B. Davey and H. Priestley. *Introduction to Lattices and Order, second edition*. Cambridge University Press, 2002.
- [DS04] Jitender S. Deogun and Jamil Saquer. Monotone concepts for formal concept analysis. *Discrete Appl. Math.*, 144(1-2) :70–78, 2004.
- [Duq99] Vincent Duquenne. Latticial structures in data analysis. *Theor. Comput. Sci.*, 217(2) :407–436, 1999.
- [EPSV98] V. Escalier, J. Pothier, Henry Soldano, and Alain Viari. Pairwise and multiple identification of three-dimensional common substructures in proteins. *Journal of Computational Biology*, 5(1) :41–56, 1998.
- [EZ05] Nahla El-Zant. *Recherche de motifs relationnels dans les séquences : Application aux séquences biologiques*. Informatique, Université Paris 13, December 2005.
- [EZO4] Nahla El-Zant and Henry Soldano. Finding repeated flexible relational words in sequences. *Journal of Systemics, Cybernetics and Informatics*, 2(4) :58–63, 2004.
- [FBDME05a] Stefano Ferilli, Teresa M. A. Basile, Nicola Di Mauro, and Floriana Esposito. Automatic induction of abduction and abstraction theories from observations. *Inductive Logic Programming*, pages 103–120, 2005.
- [FBDME05b] Stefano Ferilli, Teresa M. A. Basile, Nicola Di Mauro, and Floriana Esposito. *Automatic Induction of Abduction and Abstraction Theories from Observations*, pages 103–120. LNCS. 2005.
- [FI93] Usama M. Fayyad and Keki B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In Ruzena Bajcsy, editor, *Proceedings of the International Joint Conference on Uncertainty in AI*, pages 1022–1027. Morgan Kaufmann, 1993.
- [Für99] Johannes Fürnkranz. Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1) :3–54, February 1999.
- [Für02] Johannes Fürnkranz. A pathology of bottom-up hill-climbing in inductive rule learning. In *ALT*, volume 2533 of *LNCS*, pages 263–277. Springer, 2002.
- [FW93] Dieter Fensel and Markus Wiese. Refinement of rule sets with jojo. In P. B. Brazdil, editor, *Proceedings of the 6th European Conference on Machine Learning ECML-93, Vienna, Austria, April, 1993*, volume 667 of *Lecture Notes in Artificial Intelligence*, pages 378–383. Springer-Verlag, Berlin, 1993.

- [Gan93] J-G. Ganascia. TDIS : an algebraic formalization. In *Int. Joint Conf. on Art. Int. (IJCAI)*, volume 2, pages 1008–1013, 1993.
- [GH03] Alejandro Guerra-Hernandez. *Apprentissage d’Agents Rationnels BDI dans un univers Multi-Agents*. Informatique, Université Paris 13, December 2003.
- [GHFSS01] Alejandro Guerra-Hernández, Amal El Fallah-Seghrouchni, and Henry Soldano. Bdi multi-agent learning based on first-order induction of logical decision trees. In *International conference on Intelligent Agent Technology (IAT 2001)*, pages 160–169, Maebashi city, Japan, 2001. World Scientific Ed.
- [GHFSS04a] Alejandro Guerra-Hernández, Amal El Fallah-Seghrouchni, and Henry Soldano. Distributed learning in intentional bdi multi-agent systems. In *5th Mexican International Conference on Computer Science (ENC 2004)*, pages 225–232. IEEE Computer Society, 2004.
- [GHFSS04b] Alejandro Guerra-Hernandez, Amal El Fallah-Seghrouchni, and Henry Soldano. Learning in BDI multi-agent systems. In *Fourth International Workshop on Computational Logic in Multi-Agent Systems (CLIMA IV)*, number 3259 in LNCS, pages 218–233. Springer, 2004.
- [GK01] Bernhard Ganter and Sergei O. Kuznetsov. Pattern structures and their projections. *ICCS-01, LNCS*, 2120 :129–142, 2001.
- [GKL08] Gemma C. Garriga, Petra Kralj, and Nada Lavrač. Closed sets for labeled data. *J. Mach. Learn. Res.*, 9 :559–580, 2008.
- [GKR94] Vincent Granville, Mirko Krivánek, and Jean-Paul Rasson. Simulated annealing : A proof of convergence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(6) :652–656, 1994.
- [Glo90] F. Glover. Tabu search part i. *ORSA Journal on Computing*, 1 :190–206, 1990.
- [GRS91] Attilio Giordana, Davide Roverso, and Lorenza Saitta. Abstracting background knowledge for concept learning. In Yves Kodratoff, editor, *EWSL*, volume 482 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 1991.
- [GS90] A. Giordana and L. Saitta. Abstraction : a general framework for learning. In *Working notes of the AAAI Workshop on Automated Generation of Approximations and Abstraction, Boston, MA*, pages 245–256, 1990.
- [GSF⁺91] Attilio Giordana, Lorenza Saitta, R. Finelli, M. Paderni, and Davide Roverso. Extending inverse resolution to build up abstractions. In Edoardo Ardizzone, Salvatore Gaglio, and Filippo Sorbello, editors, *AI*IA*, volume 549 of *Lecture Notes in Computer Science*, pages 238–247. Springer, 1991.
- [GSST04] Judy Goldsmith, Robert H. Sloan, Balazs Szorenyi, and Gyorgy Turan. Theory revision with queries : Horn, read-once, and parity formulas. *Artificial Intelligence*, 156(2) :139–176, 2004.

- [GW99] B. Ganter and R. Wille. *Formal Concept Analysis : Logical Foundations*. Springer Verlag, 1999.
- [Hen94] M’hammed Henniche. Mgi : an incremental bottom-up algorithm. In *IEEE Aust. and New Zealand Conference on Intelligent Information Systems*, pages 347–351, 1994.
- [Hen97] M’hammed Henniche. Apprendre incrémentalement dans l’espace des généralisations maximales spécifiques d’instances. In *Journées Francophones d’Apprentissage, Roscoff*, pages 125–135, 1997.
- [Hir94] Haym Hirsh. Generalizing version spaces. *Mach. Learn.*, 17(1) :5–46, 1994.
- [HSS05] A. Guerra Hernandez, A. El Fallah Seghrouchni, and Henry Soldano. On learning intentionality. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 9(25) :9–18, 2005. first version published in Proceedings of IberAgents 04 Workshop.
- [JPD⁺97] Pascale Jean, Joël Pothier, Patrick M. Dansette, Daniel Mansuy, and Alain Viari. Automated multiple analysis of protein structures : Application to homology modeling of cytochromes p450 proteins : Structure, function, and genetics. *Proteins*, 28(3) :388–404, 1997.
- [KMR72] Richard M. Karp, Raymond E. Miller, and Arnold L. Rosenberg. Rapid identification of repeated patterns in strings, trees and arrays. In *Conference Record, Fourth Annual ACM Symposium on Theory of Computing*, pages 125–136, Denver, Colorado, 1–3 May 1972.
- [KR00] Antonis C. Kakas and Fabrizio Riguzzi. Abductive concept learning. *New Generation Computing*, 18(3) :243–294, 2000.
- [LAC89] A.M. Landraud, J.F. Avril, and P. Chretienne. An algorithm for finding a common structure shared by a family of strings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8) :890–895, 1989.
- [LG08] Si Q. Le and Olivier Gascuel. An improved general amino acid replacement matrix. *Mol Biol Evol*, 25(7) :1307–1320, July 2008.
- [LLSW05] Jinyan Li, Haiquan Li, Donny Soh, and Limsoon Wong. A correspondence between maximal complete bipartite subgraphs and closed patterns. *Knowledge Discovery in Databases : PKDD 2005*, pages 146–156, 2005.
- [LM90] M. Liquiere and E. MephuNguifo. Legal : un système d’apprentissage de concepts à partir d’exemples. In *Journées Francaises d’Apprentissage*, pages 93–113., 1990.
- [LS98] Michel Liquière and Jean Sallantin. Structural machine learning with galois lattice and graphs. In *ICML ’98 : Proceedings of the Fifteenth International Conference on Machine Learning*, pages 305–313, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

- [Méd91] Claudine Médigue. *Conception et Exploitation d'une Base de Données Spécialisée dédiée à l'analyse du génome d'Escherichia coli*. Sciences de la vie, Université Paris 6, December 1991.
- [Mes09] Nizar Messai. *Extraction de connaissances et Web sémantique. Application à la recherche et l'intégration de ressources génomiques sur le Web*. PhD thesis, UHP-Nancy1, Ecole Doctorale IAEM Lorraine., Mars 2009.
- [MF90] S. Muggleton and C. Feng. Efficient induction of logic programs. In *Proceedings of the 1st Conference on Algorithmic Learning Theory*, pages 368–381. Ohmsma, Tokyo, Japan, 1990.
- [Mit82] Tom M. Mitchell. Generalization as search. *Artif. Intell.*, 18(2) :203–226, 1982.
- [MKS09] Maïssa Mbaye, Francine Krief, and Henry Soldano. Collaborative strategy learning for distributed network self-configuring. In *proceedings of ComNet'09*, page to appear, November 2009.
- [Mon03] Bernard Monjardet. The presence of lattice theory in discrete problems of mathematical social sciences. why. *Mathematical Social Sciences*, 46(2) :103–144, October 2003.
- [Mug95] S. Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13(3-4) :245–286, 1995.
- [MZLS07] Jian-Min Ma, Wen-Xiu Zhang, Yee Leung, and Xiao-Xue Song. Granular computing and dual galois connection. *Inf. Sci.*, 177(23) :5365–5377, 2007.
- [NF07] Siegfried Nijssen and Elisa Fromont. Mining optimal decision trees from itemset lattices. In *KDD '07 : Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 530–539, New York, NY, USA, 2007. ACM.
- [NN97] Patrick Njiwoua and Engelbert Mephu Nguifo. Iglue : An instance-based learning system over lattice theory. In *ICTAI*, pages 75–76, 1997.
- [Oll90] Emmanuelle Ollivier. *Une méthode d'analyse de séquences biologiques indépendante du cadre de référence*. Sciences de la vie, Université Paris 6, December 1990.
- [Paw96] Zdzislaw Pawlak. Rough sets, rough relations and rough functions. *Fundamenta Informaticae*, 27(2/3) :103–108, 1996.
- [PBTL99] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1) :25–46, 1999.
- [PL05] Liviu Panait and Sean Luke. Cooperative multi-agent learning : The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3) :387–434, 2005.
- [Pol02] Lech Polkowski. *Rough Sets Mathematical Foundations*, volume 15 of *Advances in Intelligent and Soft Computing*. Springer, 2002.

- [PRSV02] Nathalie Pernelle, Marie-Christine Rousset, Henry Soldano, and Véronique Ventos. Zoom : a nested Galois lattices-based system for conceptual clustering. *J. of Experimental and Theoretical Artificial Intelligence*, 2/3(14) :157–187, 2002.
- [PRV01] Nathalie Pernelle, Marie-Christine Rousset, and Véronique Ventos. Automatic construction and refinement of a class hierarchy over multi-valued data. *Principles of Data Mining and Knowledge Discovery*, pages 386–398, 2001.
- [PSC05] N. Pisanti, Henry Soldano, and M. Carpentier. Incremental Inference of Relational Motifs with a Degenerate Alphabet. In *16th Annual Symposium on Combinatorial Pattern Matching (CPM)*, number 3537 in LNCS, pages 229–240. Springer-Verlag, 2005.
- [PSCP06] N. Pisanti, Henry Soldano, M. Carpentier, and J. Pothier. Implicit and explicit representation of approximated motifs. In C. S. Iliopoulos, K. Park, and K. Steinhoefel, editors, *Algorithms in Bioinformatics*, pages 1–15. College Press, 2006.
- [PSCP09] Nadia Pisanti, Henry Soldano, Mathilde Carpentier, and Joel Pothier. A relational extension of the notion of motifs : Application to the protein common 3d substructures searching problem. *Journal of Computational Biology*, to appear, 2009.
- [PTB⁺05] Nicolas Pasquier, Rafik Taouil, Yves Bastide, Gerd Stumme, and Lotfi Lakhal. Generating a condensed representation for association rules. *Journal Intelligent Information Systems (JIIS)*, 24(1) :29–60, 2005.
- [PYW⁺08] Jia-yang Peng, Lu-ming Yang, Jian-xin Wang, Zheng Liu, and Ming Li. An efficient algorithm for detecting closed frequent subgraphs in biological networks. In *BMEI '08 : Proceedings of the 2008 International Conference on BioMedical Engineering and Informatics*, pages 677–681, Washington, DC, USA, 2008. IEEE Computer Society.
- [RK04] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5 :101–141, January 2004.
- [Sag96] Marie-France Sagot. *Ressemblance lexicale et structurale entre macromolécules - Formalisation et approches combinatoires*. Informatique, Université de Marne-La Vallée, July 1996.
- [Sal91] Steven Salzberg. A nearest hyperrectangle learning method. *Machine Learning*, 6 :251–276, 1991.
- [SEVS95] M.-F. Sagot, V. Escalier, A. Viari, and H. Soldano. Searching for repeated words in a text allowing for mismatches and gaps. In R. Baeza-Yates and U. Manber, editors, *Second South American Workshop on String Processing*, pages 87–100, Viña del Mar, Chili, 1995. University of Chile.
- [SVC95] Henry Soldano, Alain Viari, and Marc Champesme. Searching for flexible repeated patterns using a non-transitive similarity relation. *Pattern Recognition Letters*, 16(3) :233 – 246, 1995.

- [SVPS95] M-F. Sagot, A. Viari, J. Pothier, and Henry Soldano. Finding flexible patterns in a text - an applications to 3d molecular matching. *Cabios*, 11 :59–70, 1995.
- [SVS97] M.-F. Sagot, A. Viari, and Henry Soldano. Multiple comparison : a peptide matching approach. *Theoretical Computer Science*, 180 :115–137, 1997.
- [SZ98] Lorenza Saitta and J-D Zucker. Semantic abstraction for concept representation and learning. In AAAI, editor, *Symposium on Abstraction, Reformulation and Approximation (SARA98)*, Asilomar Conference Center, Pacific Grove, California, 1998.
- [SZ07] Lorenza Saitta and Jean-Daniel Zucker. Abstraction and complexity measures. In *Abstraction, Reformulation, and Approximation*, volume 4612 of *Lecture Notes in Computer Science*, pages 375–390. Springer, 2007.
- [TNT07] Norbert Tsopzé, Engelbert Mephu Nguifo, and Gilbert Tindo. Clann : Concept lattice-based artificial neural network for supervised classification. In Peter W. Eklund, Jean Diatta, and Michel Liquiere, editors, *CLA*, volume 331 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.
- [TRS04] Alexandre Termier, Marie-Christine Rousset, and Michele Sebag. Dryade : A new approach for discovering closed frequent trees in heterogeneous tree databases. In *ICDM '04 : Proceedings of the Fourth IEEE International Conference on Data Mining*, pages 543–546, Washington, DC, USA, 2004. IEEE Computer Society.
- [UAUA04] Takeaki Uno, Tatsuya Asai, Yuzo Uchida, and Hiroki Arimura. An efficient algorithm for enumerating closed patterns in transaction databases. In *Discovery Science*, pages 16–31, 2004.
- [Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27 :1134–1142, 1984.
- [Vaz06] Vijay V. Vazirani. *Algorithmes d'approximation*. IRIS. Springer, 2006.
- [VBS01] Véronique Ventos, Pierre Brézellec, and Henry Soldano. Explicitly using default knowledge in concept learning : An extended description logics plus strict and default rules. In M. Truszczy T. Eiter, W. Faber, editor, *6th International Conference on Logic Programming and Nonmonotonic Reasoning, LPNMR 2001*, number 2173 in *Lecture Notes in Artificial Intelligence*, pages 173–185, Vienna, Austria, 2001. Springer.
- [Via04] Stéphane Vialette. On the computational complexity of 2-interval pattern matching. *Theoretical Computer Science*, 312(2–3) :223–249, 2004.
- [VM01] P. Valtchev and R. Missaoui. Building concept (Galois) lattices from parts : generalizing the incremental methods. *proceeding of ICCS-01, Lecture Notes in Computer Science*, 2120 :290–303, 2001.
- [VML02] P. Valtchev, R. Missaoui, and P. Lebrun. A partition-based approach towards building Galois (concept) lattices. *Discrete Mathematics*, 256(3) :801–829, 2002.

- [VS05a] Véronique Ventos and Henry Soldano. Alpha galois lattices : An overview. *Proceedings of ICFCA'05, Lecture Notes on Computer Science*, 3403 :298–313, 2005.
- [VS05b] Véronique Ventos and Henry Soldano. Les treillis de galois alpha. *Revue d'Intelligence Artificielle*, 19(4-5) :799–827, 2005.
- [VS08] Wolf Vanpaemel and Gert Storms. In search of abstraction : The varying abstraction model of categorization. * *WOLF VANPAEMEL* * and *GERT STORMS* *Psychonomic Bulletin and Review*, 15 :732–749, 2008.
- [VSL04a] V. Ventos, H. Soldano, and T. Lamadon. Alpha galois lattices. In *Int. Conf on Data Mining (ICDM'04)*, pages 555–558. IEEE Press, 2004.
- [VSL04b] Véronique Ventos, Henry Soldano, and T. Lamadon. Treillis de galois alpha. In *Conférence d'Apprentissage (CAP'04)*, pages 175–190, Montpellier, 2004. Presses Universitaires de Grenoble.
- [WA92] G. I. Webb. and J. Agar. Inducing diagnostic rules for glomerular disease with the dlgl machine learning algorithm. *Artificial Intelligence in Medicine*, 4(6) :419–430, 1992.
- [WL08] Lidong Wang and Xiaodong Liu. Concept analysis via rough set and afs algebra. *Inf. Sci.*, 178(21) :4125–4137, 2008.
- [WMKP06] J. Wojtusiak, R. S. Michalski, K. A. Kaufman, and J. Pietrzykowski. The aq21 natural induction program for pattern discovery : Initial version and its novel features. In *ICTAI '06 : Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, pages 523–526, Washington, DC, USA, 2006. IEEE Computer Society.
- [Yao98] Y. Y. Yao. Constructive and algebraic methods of the theory of rough sets. *Inf. Sci.*, 109(1-4) :21–47, 1998.
- [Yao04] Yiyu Yao. A partition model of granular computing. In James F. Peters, Andrzej Skowron, Jerzy W. Grzymala-Busse, Bozena Kostek, Roman W. Swiniarski, and Marcin S. Szczuka, editors, *T. Rough Sets*, volume 3100 of *Lecture Notes in Computer Science*, pages 232–253. Springer, 2004.
- [Żak83] Wojciech Żakowski. Approximations in the space (U, Π) . *Demonstratio Math.*, 16(3) :761–769, 1983.
- [Zak00] M. J. Zaki. Generating non-redundant association rules. *Intl. Conf. on Knowledge Discovery and DataMining (KDD 2000)*, 2000.
- [Zhu09] William Zhu. Relationship between generalized rough sets based on binary relation and covering. *Inf. Sci.*, 179(3) :210–225, 2009.
- [ZM09] Marco Zaffalon and Enrique Miranda. Conservative inference rule for uncertain reasoning under incompleteness. *J. Artif. Intell. Res. (JAIR)*, 34 :757–821, 2009.

- [ZO98] Mohammed J. Zaki and Mitsunori Ogihara. Theoretical foundations of association rules. In *In 3rd ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 1998.
- [Zuc03] J-D. Zucker. A grounded theory of abstraction in artificial intelligence. *Philos Trans R Soc Lond B Biol Sci*, 358(1435) :1293–309, 2003. 0962-8436 Journal Article.